Draft ETSI EN 304 635 V1.0.1 (2026-06)

HARMONISED EUROPEAN STANDARD

Cyber Security (CYBER);
CRA;
Cybersecurity requirements
for Virtualisation Execution Stack (VES)
and Container Execution Stack (CES),
including hypervisors and container runtime systems

Reference
DEN/CYBER-EUS-0016

Keywords
container, CRA, cybersecurity, virtualisation

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
ETSI Search & Browse Standards application.

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on ETSI deliver repository.

Users should be aware that the present document may be revised or have its status changed, this information is available in the Milestones listing.

If you find errors in the present document, please send your comments to the relevant service listed under Committee Support Staff.

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure (CVD) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part of this document may be reproduced in any form, by any means and in any media, without the prior written authorization of ETSI and except as expressly permitted below.
By way of exception and when the document is a normative deliverable (European Standard (EN), Technical Specification (TS), Group Specification (GS) or ETSI Standard (ES)), ETSI authorizes to reproduce and incorporate into products, services and technical documentation only those extracts (e.g. templates) that are strictly necessary for the technical implementation of the normative deliverable, to ensure compliance with the latter.
Nothing in this notice shall be construed as limiting any mandatory exceptions to copyright provided by applicable law.

© ETSI 2026.
All rights reserved.

Contents

Intellectual Property Rights 13
Foreword 13
Modal verbs terminology 14
Introduction 14
1 Scope 15
2 References 15
2.1 Normative references 15
2.2 Informative references 16
3 Definition of terms, symbols and abbreviations 17
3.1 Terms 17
3.2 Symbols 19
3.3 Abbreviations 19
4 Product Context 21
4.1 Product Functions 21
4.1.1 General 21
4.1.2 VES 21
4.1.2.1 Intended purpose and reasonably foreseeable use of the VES 21
4.1.2.2 Intended purpose and reasonably foreseeable use of the Hypervisor 21
4.1.2.3 Intended purpose and reasonably foreseeable use of the M&O system 22
4.1.3 CES 22
4.1.3.1 Intended purpose and reasonably foreseeable use of the CES 22
4.1.3.2 Intended purpose and reasonably foreseeable use of the CRS 22
4.1.3.3 Intended purpose and reasonably foreseeable use of the CE 23
4.1.3.4 Intended purpose and reasonably foreseeable use of the CO 23
4.2 Product architecture 23
4.2.1 General distinction of components 23
4.2.2 VES 23
4.2.2.1 Architectural types and virtualisation techniques of VES 23
4.2.2.2 In-scope components of the VES 24
4.2.2.2.1 Hypervisor 24
4.2.2.2.2 Management and Orchestration System 25
4.2.2.3 Security-relevant environmental dependencies (out of scope) 26
4.2.3 CES 27
4.2.3.1 In-scope components 27
4.2.3.2 Security-relevant environmental dependencies (out of scope) 28
4.2.4 Deployment of M&O, CE, and CO outside the local execution environment 28
4.2.5 Product variants and conforming products 29
4.2.5.1 Definition of conforming products 29
4.2.5.2 Product variants 29
4.3 Operational environment 30
4.3.1 General Principles 30
4.3.2 Security Objectives for the Operational Environment 30
4.3.3 Reference to Relevant CRA Harmonised Standards 32
4.4 Distribution of Security Functions 32
4.5 Users 32
4.6 Use cases 33
4.6.1 Purpose 33
4.6.2 Use Cases for VES 33
4.6.3 Use Cases for CES 35
4.6.4 Use Case to Risk Mapping 37
5 Technical Requirements for products 37
5.1 Risk-Based requirement classes/categorization and application 37
5.1.1 Requirement Classes, Categorization and Security Profiles 37

5.1.1.1 Definition of Requirement Classes: Basic, Elevated, Advanced 37
5.1.1.2 Security Profile Definitions 38
5.1.1.3 Requirement Categorization 38
5.1.1.4 Guidance for Manufacturers 39
5.1.2 Requirement Application 40
5.1.2.1 Approaches for determining security requirements 40
5.1.2.2 SP-Based Requirement Application 40
5.1.3 Applicability of requirements 41
5.2 VES Security Requirements 42
5.2.1 Hypervisor Requirements 42
5.2.1.1 Isolation 42
5.2.1.1.1 General 42
5.2.1.1.2 VM Isolation 42
5.2.1.1.3 Control Plane Isolation 43
5.2.1.1.4 Network Plane Separation 44
5.2.1.2 Integrity Protection 45
5.2.1.2.1 Boot integrity verification 45
5.2.1.2.2 Guest VM image integrity verification 46
5.2.1.2.3 Runtime integrity protection 47
5.2.1.2.4 Remote attestation 47
5.2.1.3 Authentication 48
5.2.1.3.1 General 48
5.2.1.3.2 Administrative Authentication 49
5.2.1.3.3 Service Authentication 49
5.2.1.4 Authorization 50
5.2.1.4.1 General 50
5.2.1.4.2 Administrative Authorization 50
5.2.1.4.3 Service Authorization 51
5.2.1.5 Confidentiality Protection 51
5.2.1.6 Availability and Resilience 52
5.2.1.7 Logging 53
5.2.1.8 Secure Update 54
5.2.1.9 Secure Configuration and Default 55
5.2.1.10 Data Minimisation 56
5.2.1.11 Decommissioning 56
5.2.2 Management and Orchestration System Requirements 57
5.2.2.1 General applicability 57
5.2.2.2 Authentication 57
5.2.2.3 Authorization 57
5.2.2.4 Secure Configuration 58
5.2.2.5 Communication Security 58
5.2.2.6 Integrity Protection 59
5.2.2.7 Logging 59
5.2.2.8 Secure Update 60
5.2.2.9 Decommissioning and Secure Transfer 61
5.3 CES Security Requirements 61
5.3.1 CRS Requirements 61
5.3.1.1 Isolation 61
5.3.1.1.1 Container Isolation 61
5.3.1.1.2 Control Plane Isolation 62
5.3.1.1.3 Network Plane Isolation 62
5.3.1.2 Integrity Protection 63
5.3.1.2.1 Boot integrity verification 63
5.3.1.2.2 Container image integrity verification 64
5.3.1.2.3 Runtime integrity protection 65
5.3.1.2.4 Remote attestation 65
5.3.1.3 Authentication 66
5.3.1.3.1 General 66
5.3.1.3.2 Administrative Authentication 67
5.3.1.3.3 Service Authentication 67
5.3.1.4 Authorization 68
5.3.1.4.1 General 68

5.3.1.4.2 Administrative Authorization 68
5.3.1.4.3 Service Authorization 69
5.3.1.5 Confidentiality Protection 69
5.3.1.6 Availability and Resilience 70
5.3.1.7 Logging 71
5.3.1.8 Secure Update 72
5.3.1.9 Secure Configuration and Default 73
5.3.1.10 Data Minimisation 73
5.3.1.11 Decommissioning 74
5.3.2 CE Requirements 74
5.3.2.1 Isolation 74
5.3.2.2 Integrity Protection 74
5.3.2.3 Authentication 75
5.3.2.4 Authorization 75
5.3.2.5 Confidentiality 76
5.3.2.6 Secure Update 76
5.3.2.7 Decommissioning and secure data deletion 77
5.3.3 CO Requirements 78
5.3.3.1 Isolation 78
5.3.3.2 Integrity Protection 78
5.3.3.3 Authentication 78
5.3.3.4 Authorization 79
5.3.3.5 Confidentiality 79
5.3.3.6 Secure Update 80
5.3.3.7 Decommissioning and secure data deletion 81
5.4 Assurance Security Requirements 81
5.4.1 Vulnerability Management 81
5.4.2 Software Bill of Materials (SBOM) 82

6 Assessment criteria for compliance with technical requirements 83
6.1 Assessment Methodology 83
6.1.1 General Assessment Procedure 83
6.1.2 Technology-neutral Assessment 84
6.1.2.1 General 84
6.1.2.2 Equivalence of Hardware-based and Software-based Approaches 85
6.1.2.3 Technology-neutral Assessment Principles 86
6.1.3 Assessment of RDPS requirements 87
6.2 Assessment Report Requirements 87
6.2.1 Mandatory Report Contents 87
6.2.2 Traceability Requirements 89
6.2.3 Evidence Package 89
6.3 VES 90
6.3.1 Hypervisor 90
6.3.1.1 Assessment for VM Isolation 90
6.3.1.1.1 Assessment Case AC-H-VM-ISO-001 90
6.3.1.1.2 Assessment Case AC-H-VM-ISO-002 91
6.3.1.1.3 Assessment Case AC-H-VM-ISO-003 92
6.3.1.1.4 Assessment Case AC-H-VM-ISO-004 94
6.3.1.2 Assessment for Control Plane Isolation 95
6.3.1.2.1 Assessment Case AC-H-CP-ISO-001 95
6.3.1.2.2 Assessment Case AC-H-CP-ISO-002 96
6.3.1.2.3 Assessment Case AC-H-CP-ISO-003 97
6.3.1.3 Assessment for Network Plane Separation 99
6.3.1.3.1 Assessment Case AC-H-NP-ISO-001 99
6.3.1.3.2 Assessment Case AC-H-NP-ISO-002 100
6.3.1.3.3 Assessment Case AC-H-NP-ISO-003 101
6.3.1.4 Assessment for Boot Integrity Verification 102
6.3.1.4.1 Assessment Case AC-H-B-INT-002 102
6.3.1.5 Assessment for Guest VM Image Integrity Verification 104
6.3.1.5.1 Assessment Case AC-H-IMG-INT-001 104
6.3.1.5.2 Assessment Case AC-H-IMG-INT-002 105
6.3.1.5.3 Assessment Case AC-H-IMG-INT-003 107

6.3.1.6 Assessment for Runtime Integrity Protection 108
6.3.1.6.1 Assessment Case AC-H-RP-INT-002 108
6.3.1.7 Assessment for Remote Attestation 109
6.3.1.7.1 Assessment Case AC-H-RA-INT-003 109
6.3.1.7.2 Assessment Case AC-H-RA-INT-004 111
6.3.1.8 Assessment for Administrative Authentication 113
6.3.1.8.1 Assessment Case AC-H-ADMIN-AUTH-001 113
6.3.1.8.2 Assessment Case AC-H-ADMIN-AUTH-002 114
6.3.1.8.3 Assessment Case AC-H-ADMIN-AUTH-003 115
6.3.1.9 Assessment for Service Authentication 116
6.3.1.9.1 Assessment Case AC-H-SERV-AUTH-001 116
6.3.1.9.2 Assessment Case AC-H-SERV-AUTH-002 117
6.3.1.9.3 Assessment Case AC-H-SERV-AUTH-003 118
6.3.1.10 Assessment for Administrative Authorization 119
6.3.1.10.1 Assessment Case AC-H-ADMIN-AUTHZ-001 119
6.3.1.10.2 Assessment Case AC-H-ADMIN-AUTHZ-002 120
6.3.1.10.3 Assessment Case AC-H-ADMIN-AUTHZ-003 121
6.3.1.11 Assessment for Service Authorization 122
6.3.1.11.1 Assessment Case AC-H-SERV-AUTHZ-001 122
6.3.1.11.2 Assessment Case AC-H-SERV-AUTHZ-002 123
6.3.1.11.3 Assessment Case AC-H-SERV-AUTHZ-003 124
6.3.1.12 Assessment for Confidentiality Protection 125
6.3.1.12.1 Assessment Case AC-H-CONF-001 125
6.3.1.12.2 Assessment Case AC-H-CONF-002 126
6.3.1.12.3 Assessment Case AC-H-CONF-003 128
6.3.1.12.4 Assessment Case AC-H-CONF-004 129
6.3.1.13 Assessment for Availability and Resilience 130
6.3.1.13.1 Assessment Case AC-H-AVAIL-001 130
6.3.1.13.2 Assessment Case AC-H-AVAIL-002 131
6.3.1.13.3 Assessment Case AC-H-AVAIL-003 132
6.3.1.14 Assessment for Logging 134
6.3.1.14.1 Assessment Case AC-H-LOG-001 134
6.3.1.14.2 Assessment Case AC-H-LOG-002 134
6.3.1.14.3 Assessment Case AC-H-LOG-003 135
6.3.1.15 Assessment for Secure Update 137
6.3.1.15.1 Assessment Case AC-H-UPD-001 137
6.3.1.15.2 Assessment Case AC-H-UPD-002 139
6.3.1.15.3 Assessment Case AC-H-UPD-003 140
6.3.1.16 Assessment for Secure Configuration and Default 142
6.3.1.16.1 Assessment Case AC-H-CFG-001 142
6.3.1.16.2 Assessment Case AC-H-CFG-002 143
6.3.1.16.3 Assessment Case AC-H-CFG-003 144
6.3.1.17 Assessment for Data Minimisation 145
6.3.1.17.1 Assessment Case AC-H-DM-001 145
6.3.1.17.2 Assessment Case AC-H-DM-002 146
6.3.1.18 Assessment for Decommissioning 147
6.3.1.18.1 Assessment Case AC-H-DECOM-001 147
6.3.2 M&O System 148
6.3.2.1 Assessment for Authentication 148
6.3.2.1.1 Assessment Case AC-M&O-AUTH-001 148
6.3.2.1.2 Assessment Case AC-M&O-AUTH-002 149
6.3.2.1.3 Assessment Case AC-M&O-AUTH-003 150
6.3.2.2 Assessment for Authorization 151
6.3.2.2.1 Assessment Case AC-M&O-AUTHZ-001 151
6.3.2.2.2 Assessment Case AC-M&O-AUTHZ-002 152
6.3.2.2.3 Assessment Case AC-M&O-AUTHZ-003 153
6.3.2.3 Assessment for Secure Configuration 154
6.3.2.3.1 Assessment Case AC-M&O-CFG-001 154
6.3.2.3.2 Assessment Case AC-M&O-CFG-002 155
6.3.2.3.3 Assessment Case AC-M&O-CFG-003 155
6.3.2.4 Assessment for Communication Security 156
6.3.2.4.1 Assessment Case AC-M&O-COM-001 156

6.3.2.4.2 Assessment Case AC-M&O-COM-002 157
6.3.2.4.3 Assessment Case AC-M&O-COM-003 158
6.3.2.5 Assessment for Integrity Protection 159
6.3.2.5.1 Assessment Case AC-M&O-INT-001 159
6.3.2.5.2 Assessment Case AC-M&O-INT-002 160
6.3.2.5.3 Assessment Case AC-M&O-INT-003 161
6.3.2.6 Assessment for Logging 162
6.3.2.6.1 Assessment Case AC-M&O-LOG-001 162
6.3.2.6.2 Assessment Case AC-M&O-LOG-002 162
6.3.2.6.3 Assessment Case AC-M&O-LOG-003 163
6.3.2.7 Assessment for Secure Update 164
6.3.2.7.1 Assessment Case AC-M&O-UPD-001 164
6.3.2.7.2 Assessment Case AC-M&O-UPD-002 166
6.3.2.7.3 Assessment Case AC-M&O-UPD-003 168
6.3.2.8 Assessment for Decommissioning and Secure Transfer 169
6.3.2.8.1 Assessment Case AC-M&O-DECOM-001 169
6.3.2.8.2 Assessment Case AC-M&O-DECOM-002 170
6.4 CES 171
6.4.1 CRS 171
6.4.1.1 Assessment for Container Isolation 171
6.4.1.1.1 Assessment Case AC-CRS-CN-ISO-001 171
6.4.1.1.2 Assessment Case AC-CRS-CN-ISO-002 172
6.4.1.1.3 Assessment Case AC-CRS-CN-ISO-003 174
6.4.1.2 Assessment for Control Plane Isolation 175
6.4.1.2.1 Assessment Case AC-CRS-CP-ISO-001 175
6.4.1.2.2 Assessment Case AC-CRS-CP-ISO-002 176
6.4.1.2.3 Assessment Case AC-CRS-CP-ISO-003 177
6.4.1.3 Assessment for Network Plane Isolation 179
6.4.1.3.1 Assessment Case AC-CRS-NP-ISO-001 179
6.4.1.3.2 Assessment Case AC-CRS-NP-ISO-002 180
6.4.1.3.3 Assessment Case AC-CRS-NP-ISO-003 181
6.4.1.4 Assessment for Boot Integrity Verification 183
6.4.1.4.1 Assessment Case AC-CRS-B-INT-002 183
6.4.1.5 Assessment for Container Image Integrity Verification 185
6.4.1.5.1 Assessment Case AC-CRS-IMG-INT-001 185
6.4.1.5.2 Assessment Case AC-CRS-IMG-INT-002 186
6.4.1.5.3 Assessment Case AC-CRS-IMG-INT-003 187
6.4.1.6 Assessment for Runtime Integrity Protection 189
6.4.1.6.1 Assessment Case AC-CRS-RP-INT-002 189
6.4.1.7 Assessment for Remote Attestation 190
6.4.1.7.1 Assessment Case AC-CRS-RA-INT-003 190
6.4.1.7.2 Assessment Case AC-CRS-RA-INT-004 192
6.4.1.8 Assessment for Administrative Authentication 194
6.4.1.8.1 Assessment Case AC-CRS-ADMIN-AUTH-001 194
6.4.1.8.2 Assessment Case AC-CRS-ADMIN-AUTH-002 195
6.4.1.8.3 Assessment Case AC-CRS-ADMIN-AUTH-003 196
6.4.1.9 Assessment for Service Authentication 197
6.4.1.9.1 Assessment Case AC-CRS-SERV-AUTH-001 197
6.4.1.9.2 Assessment Case AC-CRS-SERV-AUTH-002 198
6.4.1.9.3 Assessment Case AC-CRS-SERV-AUTH-003 199
6.4.1.10 Assessment for Administrative Authorization 200
6.4.1.10.1 Assessment Case AC-CRS-ADMIN-AUTHZ-001 200
6.4.1.10.2 Assessment Case AC-CRS-ADMIN-AUTHZ-002 201
6.4.1.10.3 Assessment Case AC-CRS-ADMIN-AUTHZ-003 202
6.4.1.11 Assessment for Service Authorization 203
6.4.1.11.1 Assessment Case AC-CRS-SERV-AUTHZ-001 203
6.4.1.11.2 Assessment Case AC-CRS-SERV-AUTHZ-002 204
6.4.1.11.3 Assessment Case AC-CRS-SERV-AUTHZ-003 205
6.4.1.12 Assessment for Confidentiality Protection 206
6.4.1.12.1 Assessment Case AC-CRS-CONF-001 206
6.4.1.12.2 Assessment Case AC-CRS-CONF-002 208
6.4.1.12.3 Assessment Case AC-CRS-CONF-003 209

6.4.1.12.4 Assessment Case AC-CRS-CONF-004 211
6.4.1.13 Assessment for Availability and Resilience 213
6.4.1.13.1 Assessment Case AC-CRS-AVAIL-001 213
6.4.1.13.2 Assessment Case AC-CRS-AVAIL-002 214
6.4.1.13.3 Assessment Case AC-CRS-AVAIL-003 215
6.4.1.13.4 Assessment Case AC-CRS-AVAIL-004 216
6.4.1.14 Assessment for Logging 217
6.4.1.14.1 Assessment Case AC-CRS-LOG-001 217
6.4.1.14.2 Assessment Case AC-CRS-LOG-002 218
6.4.1.14.3 Assessment Case AC-CRS-LOG-003 219
6.4.1.15 Assessment for Secure Update 220
6.4.1.15.1 Assessment Case AC-CRS-UPD-001 220
6.4.1.15.2 Assessment Case AC-CRS-UPD-002 222
6.4.1.15.3 Assessment Case AC-CRS-UPD-003 224
6.4.1.16 Assessment for Secure Configuration and Default 225
6.4.1.16.1 Assessment Case AC-CRS-CFG-001 225
6.4.1.16.2 Assessment Case AC-CRS-CFG-002 226
6.4.1.16.3 Assessment Case AC-CRS-CFG-003 227
6.4.1.17 Assessment for Data Minimisation 228
6.4.1.17.1 Assessment Case AC-CRS-DM-001 228
6.4.1.17.2 Assessment Case AC-CRS-DM-002 229
6.4.1.18 Assessment for Decommissioning 230
6.4.1.18.1 Assessment Case AC-CRS-DECOM-001 230
6.4.2 CE 231
6.4.2.1 Assessment for Isolation 231
6.4.2.1.1 Assessment Case AC-CE-ISO-001 231
6.4.2.1.2 Assessment Case AC-CE-ISO-002 232
6.4.2.1.3 Assessment Case AC-CE-ISO-003 233
6.4.2.2 Assessment for Integrity Protection 234
6.4.2.2.1 Assessment Case AC-CE-INTEG-001 234
6.4.2.2.2 Assessment Case AC-CE-INTEG-002 235
6.4.2.2.3 Assessment Case AC-CE-INTEG-003 236
6.4.2.3 Assessment for Authentication 237
6.4.2.3.1 Assessment Case AC-CE-AUTH-001 237
6.4.2.3.2 Assessment Case AC-CE-AUTH-002 238
6.4.2.3.3 Assessment Case AC-CE-AUTH-003 238
6.4.2.4 Assessment for Authorization 239
6.4.2.4.1 Assessment Case AC-CE-AUTHZ-001 239
6.4.2.4.2 Assessment Case AC-CE-AUTHZ-002 240
6.4.2.4.3 Assessment Case AC-CE-AUTHZ-003 241
6.4.2.5 Assessment for Confidentiality and Data Protection 242
6.4.2.5.1 Assessment Case AC-CE-CONF-001 242
6.4.2.5.2 Assessment Case AC-CE-CONF-002 243
6.4.2.5.3 Assessment Case AC-CE-CONF-003 243
6.4.2.6 Assessment for Secure Update 245
6.4.2.6.1 Assessment Case AC-CE-UPD-001 245
6.4.2.6.2 Assessment Case AC-CE-UPD-002 247
6.4.2.7 Assessment for Decommissioning and Secure Transfer 248
6.4.2.7.1 Assessment Case AC-CE-DECOM-001 248
6.4.2.7.2 Assessment Case AC-CE-DECOM-002 249
6.4.3 CO 250
6.4.3.1 Assessment for Isolation 250
6.4.3.1.1 Assessment Case AC-CO-ISO-001 250
6.4.3.1.2 Assessment Case AC-CO-ISO-002 251
6.4.3.1.3 Assessment Case AC-CO-ISO-003 252
6.4.3.2 Assessment for Integrity Protection 252
6.4.3.2.1 Assessment Case AC-CO-INTEG-001 252
6.4.3.2.2 Assessment Case AC-CO-INTEG-002 253
6.4.3.2.3 Assessment Case AC-CO-INTEG-003 254
6.4.3.3 Assessment for Authentication 255
6.4.3.3.1 Assessment Case AC-CO-AUTH-001 255
6.4.3.3.2 Assessment Case AC-CO-AUTH-002 256

6.4.3.3.3 Assessment Case AC-CO-AUTH-003 256
6.4.3.4 Assessment for Authorization 257
6.4.3.4.1 Assessment Case AC-CO-AUTHZ-001 257
6.4.3.4.2 Assessment Case AC-CO-AUTHZ-002 258
6.4.3.4.3 Assessment Case AC-CO-AUTHZ-003 259
6.4.3.5 Assessment for Confidentiality and Data Protection 260
6.4.3.5.1 Assessment Case AC-CO-CONF-001 260
6.4.3.5.2 Assessment Case AC-CO-CONF-002 261
6.4.3.5.3 Assessment Case AC-CO-CONF-003 261
6.4.3.6 Assessment for Secure Update 262
6.4.3.6.1 Assessment Case AC-CO-UPD-001 262
6.4.3.6.2 Assessment Case AC-CO-UPD-002 264
6.4.3.6.3 Assessment Case AC-CO-UPD-003 266
6.4.3.7 Assessment for Decommissioning and Secure Transfer 267
6.4.3.7.1 Assessment Case AC-CO-DECOM-001 267
6.4.3.7.2 Assessment Case AC-CO-DECOM-002 268
6.5 Assessment for Assurance Security Requirements 269
6.5.1 Assessment for Vulnerability Management 269
6.5.1.1 Assessment Case AC-ASSUR-VULM-001 269
6.5.2 Assessment for Software Bill of Materials (SBOM) 270
6.5.2.1 Assessment Case AC-ASSUR-SBOM-001 270
6.5.2.2 Assessment Case AC-ASSUR-SBOM-002 271
6.5.2.3 Assessment Case AC-ASSUR-SBOM-003 273

Annex A (informative): Relationship between the present document and the requirements of EU Regulation (EU) 2024/2847 - the Cyber Resilience Act 274

Annex B (informative): Security Analysis 278

B.1 Overview 278

B.2 Threats 278
B.2.1 Overview 278
B.2.2 VES 279
B.2.2.1 Hyper Type I 279
B.2.2.2 Hyper Type II 280
B.2.2.3 Orchestration and Management system 281
B.2.3 CES 282
B.2.3.1 Container Orchestrator 282
B.2.3.2 Container Engine 284
B.2.3.3 Container Runtime System 285
B.2.4 Hardware-level common threat 286

B.3 Risk Assessment Process 286

B.4 Risk Factor Scoring 287

B.5 Calculation and Mapping to Likelihood/Impact Levels 290
B.5.1 General 290
B.5.2 Likelihood Score 290
B.5.3 Impact Score 290

B.6 Risk Matrix 291

B.7 Risk Methodology Applied to Use Cases 291
B.7.1 General 291
B.7.2 Risk Evaluation of VES use cases 292
B.7.2.1 Hypervisor Type I 292
B.7.2.2 Hypervisor Type II 296
B.7.2.3 M&O System 300
B.7.3 Risk Evaluation of CES use cases 304
B.7.3.1 CRS 304
B.7.3.2 CE 309
B.7.3.3 CO 313

B.7.4 Risk Evaluation of common hardware-level threat T-ALL-HW-ACCESS....318
B.8 Threat-to-Requirement Traceability Matrix....321
B.8.1 VES....321
B.8.1.1 Hypervisor....321
B.8.1.2 M&O System....322
B.8.2 CES....323
B.8.2.1 CRS....323
B.8.2.2 CE....324
B.8.2.3 CO....324
B.8.3 Hardware-level common threat....325
Annexes C to J: Void....326
Annex K (normative): Generic cryptographic requirements and assessment....327
K.1 Cryptography....327
K.1.1 Requirement....327
K.1.2 Assessment of product cryptographic configuration....328
K.1.2.0 General....328
K.1.2.1 Assessment of ACM-listed cryptographic mechanisms....328
K.1.2.1.1 Assessment objective....328
K.1.2.1.2 Assessment preparation....328
K.1.2.1.3 Assessment activities....328
K.1.2.1.4 Assessment evidence....329
K.1.2.1.5 Assessment verdict....329
K.1.2.2 Assessment of ACM-extended cryptographic mechanisms....329
K.1.2.2.1 Assessment objective....329
K.1.2.2.2 Assessment preparation....329
K.1.2.2.3 Assessment activities....329
K.1.2.2.4 Assessment evidence....330
K.1.2.2.5 Assessment verdict....330
K.1.2.3 Assessment of interoperability-based cryptographic mechanisms....330
K.1.2.3.1 Assessment objective....330
K.1.2.3.2 Assessment preparation....330
K.1.2.3.3 Assessment activities....330
K.1.2.3.4 Assessment evidence....331
K.1.2.3.5 Assessment verdict....331
K.2 Crypto agility....331
K.2.1 Requirement....331
K.2.2 Assessment of crypto-agility....332
K.2.2.1 Assessment objective....332
K.2.2.2 Assessment preparation....332
K.2.2.3 Assessment activities....333
K.2.2.4 Assessment evidence....333
K.2.2.5 Assessment verdict....333
K.3 ACM-extended cryptographic mechanisms....333
K.3.1 Requirement....333
K.3.2 List of ACM-extended cryptographic mechanisms....334
K.3.2.1 Stream cipher mechanisms....334
K.3.2.1.1 ChaCha20....334
K.3.2.2 Authenticated encryption mechanisms....334
K.3.2.2.1 ChaCha20-Poly1305....334
K.3.2.3 Message authentication mechanisms....335
K.3.2.3.1 Poly1305....335
K.3.2.4 Password protection mechanisms....335
K.3.2.4.1 Argon2id....335
K.3.2.5 Elliptic-curve key agreement mechanisms....336
K.3.2.5.1 X25519 and X448....336
K.3.2.6 Digital signature mechanisms....336
K.3.2.6.1 Ed25519 and Ed448....336

K.3.2.7 Container image, artefact, attestation and metadata protection formats 337
K.3.2.7.1 COSE 337
K.3.2.7.2 JOSE / JWS / JWE 337
K.3.2.8 Protected communication profiles 338
K.3.2.8.1 QUIC 338
K.3.2.9 Secure time protocols 338
K.3.2.9.1 Network Time Security for NTP 338
K.3.2.10 Secure device and platform identity 339
K.3.2.10.1 IEEE 802.1AR secure device identity 339
K.3.3 Assessment 340

K.4 Interoperability-based cryptographic mechanisms 340
K.4.1 Requirement 340
K.4.2 List of interoperability-based cryptographic mechanisms 340
K.4.2.1 Password protection mechanisms 340
K.4.2.1.1 scrypt 340
K.4.3 Assessment 340

Annexes L to Q: Void 341

Annex R (normative): Additional provisions for products relying on Remote Data Processing Solutions (RDPS) 342

R.1 Scope & Applicability 342

R.2 RDPS as a product-boundary extension 342

R.3 Threat Model 343
R.3.0 General 343
R.3.1 Assets 344
R.3.2 Threat catalogue 344
R.3.3 Assets ↔ Threats mapping 345

R.4 Security Requirements 345
R.4.1 General 345
R.4.2 Applicability of Annex R requirements 345
R.4.3 Local product side requirements 346
R.4.3.1 Data authenticity of interactions received from the RDPS side 346
R.4.3.2 Integrity of interactions received from the RDPS side 346
R.4.3.3 Confidentiality of data exchanged with the RDPS side 347
R.4.3.4 Availability 347
R.4.4 RDPS side requirements 348
R.4.4.1 Data authenticity of interactions received from the local product side 348
R.4.4.2 Integrity of interactions received from the local product side 349
R.4.4.3 Confidentiality of data exchanged with the local product side 349
R.4.4.4 Availability 350
R.4.5 Threats ↔ Requirements mapping 351

R.5 Conformity assessment 351
R.5.0 General 351
R.5.1 Local product side requirements 351
R.5.1.1 Assessment case for REQ-RDPS-L-AUTH-001 351
R.5.1.2 Assessment case for REQ-RDPS-L-AUTH-002 352
R.5.1.3 Assessment case for REQ-RDPS-L-AUTH-003 353
R.5.1.4 Assessment case for REQ-RDPS-L-INTEG-001 354
R.5.1.5 Assessment case for REQ-RDPS-L-INTEG-002 355
R.5.1.6 Assessment case for REQ-RDPS-L-INTEG-003 356
R.5.1.7 Assessment case for REQ-RDPS-L-CONF-001 356
R.5.1.8 Assessment case for REQ-RDPS-L-CONF-002 357
R.5.1.9 Assessment case for REQ-RDPS-L-CONF-003 358
R.5.1.10 Assessment case for REQ-RDPS-L-AVAIL-001 359
R.5.1.11 Assessment case for REQ-RDPS-L-AVAIL-002 360
R.5.1.12 Assessment case for REQ-RDPS-L-AVAIL-003 361
R.5.2 RDPS side requirements 362

R.5.2.1 Assessment case for REQ-RDPS-R-AUTH-001 362
R.5.2.2 Assessment case for REQ-RDPS-R-AUTH-002 362
R.5.2.3 Assessment case for REQ-RDPS-R-AUTH-003 363
R.5.2.4 Assessment case for REQ-RDPS-R-INTEG-001 364
R.5.2.5 Assessment case for REQ-RDPS-R-INTEG-002 365
R.5.2.6 Assessment case for REQ-RDPS-R-INTEG-003 366
R.5.2.7 Assessment case for REQ-RDPS-R-CONF-001 367
R.5.2.8 Assessment case for REQ-RDPS-R-CONF-002 368
R.5.2.9 Assessment case for REQ-RDPS-R-CONF-003 368
R.5.2.10 Assessment case for REQ-RDPS-R-AVAIL-001 369
R.5.2.11 Assessment case for REQ-RDPS-R-AVAIL-002 370
R.5.2.12 Assessment case for REQ-RDPS-R-AVAIL-003 371

Annexes S to X: Void 373

Annex Y (informative): Mapping of CES/VES cryptographic requirements to ACM and Annex K mechanism families 374

Y.1 General 374

Y.2 Mapping of Hypervisor requirements 374

Y.3 Mapping of Management and Orchestration requirements 376

Y.4 Mapping of CRS requirements 377

Y.5 Mapping of Container Engine requirements 379

Y.6 Mapping of Container Orchestrator requirements 381

Annex Z (informative): Change history 382

History 386

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables (European Standard (EN), Technical Specification (TS), Group Specification (GS) or ETSI Standard (ES)) may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI IPR online database.

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This draft Harmonised European Standard (EN) has been produced by ETSI Technical Committee Cyber Security (CYBER), and is now submitted for the combined Public Enquiry and Vote phase of the ETSI Standardisation Request deliverable Approval Procedure (SRdAP).

The present document has been prepared under the Commission's standardisation request C(2025) 618 [i.3] final to provide one voluntary means of conforming to the requirements of Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) No 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) (CRA) [i.1].

Once the present document is cited in the Official Journal of the European Union under that Regulation, compliance with the normative clauses of the present document given in Table A.1 confers, within the limits of the scope of the present document, a presumption of conformity with the corresponding requirements of that Regulation and associated EFTA regulations.

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	18 months after doa

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Introduction

The present document defines cybersecurity requirements applicable to Virtualisation Execution Stack (VES) and Container Execution Stack (CES), including hypervisors and container runtime systems. It applies to products with digital elements designed to host, manage, and isolate virtualised workloads and execution environments, using:

- Virtual machines (VMs): full system emulation with independent operating systems; or
- Containers: process isolation using OS-level mechanisms.

The present document supports compliance with Regulation (EU) 2024/2847, the Cyber Resilience Act, specifically addressing the essential cybersecurity requirements defined in Annex I, Parts I and II.

The present document is intended to be applied as follows:

1) the manufacturer identifies whether the product is a VES or a CES, and determines which components are included within the product boundary in accordance with clauses 4.1 and 4.2;
2) the manufacturer identifies the relevant product use case in accordance with clause 4.6 and associated risk level in accordance with clause 4.6.4 and, where needed, Annex B;
3) the manufacturer determines the applicable Security Profile (SP) and the corresponding requirement application logic in accordance with clause 5;
4) the manufacturer applies only those requirements specified in clause 5 that are relevant to the components included in the product, its intended use, and the applicable SP; and
5) conformity assessment is performed against the corresponding assessment provisions in clause 6.

1 Scope

The present document specifies technical requirements and corresponding assessment criteria for Virtualisation Execution Stack (VES) and Container Execution Stack (CES) products, including hypervisors and container runtime systems, related to cybersecurity. The products with digital elements in scope, thereafter referred to as the "product":

- are specified within the "technical description" of the "category of product" in Class II, point 1 by the Commission Implementing Regulation (EU) 2025/2392 [i.2] as:
  "Hypervisors and container runtime systems that support virtualised execution of operating systems and similar environments";
- are only covered within the product context described in clause 4.

The present document covers those products to demonstrate compliance with essential cybersecurity requirements in the Regulation (EU) 2024/2847 [i.1], Annex I, Part I under the conditions identified in Annex A.

Commission Implementing Regulation (EU) 2025/2392 [i.2] identifies hypervisors and container runtime systems as core components. However, actual market products typically include additional elements beyond the hypervisor kernel or container runtime binary. These additional components provide essential management, orchestration, and operational capabilities that are necessary for real-world deployment and are therefore included within the scope of the present document.

The present document addresses the CRA Class II, point 1 product category within the following product contexts:

- Virtualisation Execution Stack (VES) for hypervisor-based environments; and
- Container Execution Stack (CES) for container-based environments.

The corresponding terms and definitions are provided in clause 3. The architectural decomposition, in-scope components, and security-relevant environmental dependencies are specified in clause 4.

Accordingly, the present document defines security requirements not only for the core execution systems identified in the CRA but also for the broader product context in which these systems are deployed, ensuring alignment with market reality and comprehensive coverage of security risks. The Management and Orchestration (M&O) System, Container Engine (CE), and Container Orchestrator (CO) are covered by the present document and are in scope only where they are developed or provided by the manufacturer, or under the responsibility of the manufacturer, as part of the declared product.

Any usage of AI agents is out of scope of the present document.

Where the product includes or depends on components that are outside the scope of the present document, the applicable requirements are to be addressed through the relevant operational-environment provisions or other relevant harmonised standards, as identified in clause 4.3.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents are necessary for the application of the present document.

[1] ENISA-ECCG: "Agreed Cryptographic Mechanisms", Version 2.0, April 2025.

[2] IRTF RFC 8439 (June 2018): "ChaCha20 and Poly1305 for IETF Protocols".

[3] IRTF RFC 9106 (September 2021): "Argon2 Memory-Hard Function for Password Hashing and Proof-of-Work Applications".

[4] IETF RFC 7914 (August 2016): "The scrypt Password-Based Key Derivation Function".

[5] IRTF RFC 7748 (January 2016): "Elliptic Curves for Security".

[6] IETF RFC 8410 (August 2018): "Algorithm Identifiers for Ed25519, Ed448, X25519, and X448 for Use in the Internet X.509 Public Key Infrastructure".

[7] IRTF RFC 8032 (January 2017): "Edwards-Curve Digital Signature Algorithm (EdDSA)".

[8] IETF RFC 9052 (August 2022): "CBOR Object Signing and Encryption (COSE): Structures and Process".

[9] IETF RFC 9053 (August 2022): "CBOR Object Signing and Encryption (COSE): Initial Algorithms".

[10] IETF RFC 7515 (May 2015): "JSON Web Signature (JWS)".

[11] IETF RFC 7516 (May 2015): "JSON Web Encryption (JWE)".

[12] IETF RFC 9000 (May 2021): "QUIC: A UDP-Based Multiplexed and Secure Transport".

[13] IETF RFC 9001 (May 2021): "Using TLS to Secure QUIC".

[14] IRTF RFC 8915 (September 2020): "Network Time Security for the Network Time Protocol".

[15] IEEE 802.1AR™-2018: "IEEE Standard for Local and Metropolitan Area Networks - Secure Device Identity".

[16] ETSI TS 119 312 (V1.5.1) (2024-12): "Electronic Signatures and Trust Infrastructures (ESI); Cryptographic Suites".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long-term validity.

The following referenced documents may be useful in implementing an ETSI deliverable or add to the reader's understanding but are not required for conformance to the present document.

[i.1] Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act).

[i.2] Commission Implementing Regulation (EU) 2025/2392 of 28 November 2025 on the technical description of the categories of important and critical products with digital elements pursuant to Regulation (EU) 2024/2847 of the European Parliament and of the Council.

[i.3] C(2025)618 - Standardisation request M/606: Commission Implementing decision of 3.2.2025 on a standardisation request to the European Committee for Standardisation (CEN), the European Committee for Electrotechnical Standardisation (Cenelec) and the European Telecommunications Standards Institute (ETSI) as regards products with digital elements in support of Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020and Directive (EU) 2020/1828 (Cyber Resilience Act).

[i.4] ETSI EN 304 623: "Cyber Security (CYBER); CRA; Cybersecurity requirements for boot managers".

[i.5] ETSI EN 304 626: "Cyber Security (CYBER); CRA; Cybersecurity requirements for operating systems".

[i.6] ETSI EN 304 625: "Cyber Security (CYBER); CRA; Cybersecurity requirements for physical and virtual network interfaces".

[i.7] ETSI EN 304 624: "Cyber Security (CYBER); CRA; Cybersecurity requirements for Public key infrastructure and digital certificate issuance software".

[i.8] ETSI EN 304 622: "Cyber Security (CYBER); CRA; Cybersecurity requirements for Security information and event management (SIEM) systems".

[i.9] ETSI EN 304 636: "Cyber Security (CYBER); CRA; Cybersecurity requirements for firewalls, intrusion detection and/or prevention systems".

[i.10] prEN 40000-1-1: "Cybersecurity requirements for products with digital elements - Vocabulary".

[i.11] prEN 40000-1-2: "Cybersecurity requirements for products with digital elements - Part 1-2: Principles for cyber resilience".

[i.12] NIST FIPS 186-5: "Digital Signature Standard (DSS)".

[i.13] prEN 40000-1-3: "Cybersecurity requirements for products with digital elements - Vulnerability Handling".

[i.14] BSI TR-02102-1 (Version 2026-01, 31 January 2026): "Cryptographic Mechanisms: Recommendations and Key Lengths".

[i.15] BSI TR-02102-2 (Version 2026-01): "Cryptographic Mechanisms: Recommendations and Key Lengths - Use of Transport Layer Security (TLS)".

[i.16] BSI TR-02102-3 (Version 2026-01): "Cryptographic Mechanisms: Recommendations and Key Lengths - Use of Internet Protocol Security (IPsec) and Internet Key Exchange (IKEv2)".

[i.17] BSI TR-02102-4 (Version 2026-01): "Cryptographic Mechanisms: Recommendations and Key Lengths - Use of Secure Shell (SSH)".

[i.18] ANSSI Référentiel Général de Sécurité, Annex B2, version 2.0: "Gestion des clés cryptographiques - Règles et recommandations concernant la gestion des clés utilisées dans les mécanismes cryptographiques".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in CRA Regulation [i.1], prEN 40000-1-1 [i.10] and the following apply.

confidential computing: security capability that provides data-in-use protection through hardware-based isolated execution

container: software-based isolated execution environment that packages an application with its dependencies and runs using the host operating system kernel

Container Engine (CE): software component that manages the lifecycle of containers on a host, including image handling, container creation, execution, logging, storage, and related management interfaces

Container Execution Stack (CES): layered set of software components enabling the deployment, management, and execution of containerized applications

Container Orchestrator (CO): software component that automates the deployment, scheduling, scaling, networking, availability, and lifecycle management of containerized applications across one or more machines

dedicated management network / interface: network interface or network segment that is exclusively used for management plane traffic and is not shared with guest workload traffic, storage traffic, data-plane traffic, VM/container traffic, or tenant-accessible networks

NOTE : A dedicated management interface may be implemented as:

- a physically separate network interface assigned solely to management traffic; or
- a logically isolated network (e.g. VLAN, VXLAN, VRF) where isolation is enforced by the hypervisor / CRS / host networking layer and cannot be modified or bypassed by tenant workloads.

domain: separated part of a product used to group components, functions, workloads, users, or interfaces that are managed or protected together

NOTE 1: A domain can be a management domain, workload domain, tenant domain, driver domain, host domain, network domain, or security domain.

NOTE 2: A domain may be implemented using different mechanisms, including virtualisation, container isolation, namespaces, access-control policies, network segmentation, privilege separation, or policy enforcement.

NOTE 3: In the present document, "domain" is a generic architecture-neutral term and does not imply a specific technology or product-specific construct.

hardware-based root of trust: hardware-based trust anchor providing security functions used to establish or maintain trust in the product, such as protected key storage, measured boot, verified boot, attestation support, or cryptographic operations

NOTE: Examples may include Trusted Platform Modules (TPM), secure elements, or equivalent platform trust anchors.

hardware-mediated execution enclave: execution environment in which hardware mechanisms provide protected isolation for code and data during execution and control access to memory and execution state from software outside that environment

NOTE: Examples may include trusted execution environments or equivalent platform mechanisms providing isolated execution.

hardware security module: hardware or hardware-assisted component providing dedicated security functions, such as protected storage of keys or credentials, cryptographic processing, attestation support, or resistance against unauthorized access or modification

NOTE: Examples may include Hardware Security Modules (HSM), Trusted Platform Modules (TPM), secure elements, or equivalent hardware security components.

hybrid hypervisor: hypervisor architecture that combines characteristics of bare-metal and hosted hypervisors, where the hypervisor relies on a general-purpose host operating system, privileged virtualised execution environment, management domain, driver domain, or equivalent component for device drivers, management, or orchestration functions

hypervisor: software that enables the creation and management of virtual machines by abstracting, allocating and controlling access to a host's hardware resources, providing isolation between virtual machines

impact: severity of harm if a threat event occurs

likelihood: probability of a threat event occurring

microVM: lightweight virtual machine designed for minimal resource overhead and rapid startup, providing an isolated execution environment with a reduced attack surface compared to traditional virtual machines

multi-tenancy: architectural property of a virtualised or containerized environment in which multiple tenants share the same underlying infrastructure while maintaining isolation of their workloads and data

paravirtualisation: virtualisation approach where the guest OS is modified to make direct API calls (hypercalls) to the hypervisor instead of issuing hardware instructions that would otherwise be intercepted and emulated

paravisor: guest-resident virtualisation support component that runs at a higher virtual machine privilege level than the guest operating system within the virtualised computing environment of a virtual machine and provides virtualisation services or compatibility functions to the guest operating system from within that virtualised computing environment

NOTE: A paravisor may be loaded, authorized, constrained, or transitioned to by the Hypervisor or by another component within the declared product architecture.

strong authentication: authentication that provides high-assurance identity verification for access to administrative interfaces, control functions or sensitive data, implemented using one or more mechanisms such as Multi-Factor Authentication (MFA), certificate-based authentication, cryptographic key-based authentication using protected private keys, hardware-backed authentication mechanisms, or equivalent mechanisms providing the same assurance

tenant: isolated unit of use and management within a virtualised or containerized environment, representing a single user or organization sharing the underlying infrastructure

type I hypervisor (Bare-Metal / Native): hypervisor that runs directly on physical hardware to create and manage virtual machines

type II hypervisor (Hosted): hypervisor that runs as an application on top of a general-purpose host operating system to create and manage virtual machines

Virtualisation Execution Stack (VES): set of software components responsible for deploying, running, and managing Virtual Machines (VMs) on shared physical infrastructure

Virtual Machine (VM): software-based virtualised computing environment whose primary purpose is to run a guest operating system on abstracted hardware resources provided by a hypervisor

Virtual Machine Privilege Level (VMPL): mode of execution that may exist within the virtualised computing environment of a virtual machine and operate on a per-virtual-machine or per-virtual-processor basis, where software running at a higher virtual machine privilege level may be protected from software running at a lower virtual machine privilege level through hardware-provided virtualisation extensions or equivalent mechanisms

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ABAC	Attribute-Based Access Control
ACL	Access Control List
ACM	Agreed Cryptographic Mechanisms
AES	Advanced Encryption Standard
API	Application Programming Interface
ASLR	Address Space Layout Randomization
ASSUR	Assurance
BIOS	Basic Input/Output System
CBOR	Concise Binary Object Representation
CE	Container Engine
CES	Container Execution Stack
CFG	Configuration
CI/CD	Continuous Integration / Continuous Delivery
CLI	Command Line Interface
CO	Container Orchestrator
CONF	Confidentiality
COSE	CBOR Object Signing and Encryption
CRA	Cyber Resilience Act
CRI-O	Container Runtime Interface - Open Container Initiative

CRS	Container Runtime System
CPU	Central Processing Unit
CTL	Control
DoS	Denial-of-Service
EdDSA	Edwards-curve Digital Signature Algorithm
ESXi	Elastic Sky X tntegrated (VMware hypervisor name)
GCM	Galois/Counter Mode (for encryption)
gRPC	gRPC Remote Procedure Call
HSM	Hardware Security Module
IaaS	Infrastructure as a Service
I/O	Input/Output
ID	Identification
IMG	Image
IOMMU	Input/Output Memory Management Unit
ISO	International Organization for Standardization
IT	Information Technology
JOSE	JavaScript Object Signing and Encryption
JSON	JavaScript Object Notation
LOG	Logging
M&O	Management and Orchestration
MFA	Multi-Factor Authentication
mTLS	mutual Transport Layer Security
NIC	Network Interface Card
NTP	Network Time Protocol
OE	Operational Environment
OS	Operating System
OSS	Open Source Software
PC	Personal Computer
PDP	Policy Decision Point
PKI	Public Key Infrastructure
RAM	Random Access Memory
RBAC	Role-Based Access Control
REQ	Requirement
SBOM	Software Bill Of Materials
SDN	Software-Defined Networking
SIEM	Security Information and Event Management
SP	Security Profile
SPIFFE	Secure Production Identity Framework Eor Everyone
SR-IOV	Single Root Input/Output Virtualisation
SUP	Support
TLS	Transport Layer Security
TPM	Trusted Platform Module
TS	Technical Specification
U/C	Use Case
UEFI	Unified Extensible Firmware Interface
UI	User Interface
UPD	Update
VES	Virtualisation Execution Stack
VLAN	Virtual Local Area Network
VM	Virtual Machine
VMM	Virtual Machine Monitor (Hypervisor Core)
VPN	Virtual Private Network
vTPM	virtual Trusted Platform Module
VULM	Vulnerability Management
VXLAN	Virtual Extensible LAN

4 Product Context

4.1 Product Functions

4.1.1 General

The present document applies to products whose intended purpose is to provide virtualisation execution or container execution functions, including workload execution, workload isolation, lifecycle management, configuration, administration, and, where applicable, orchestration and policy enforcement.

Reasonably foreseeable use includes operation in local, remote, clustered, distributed, or externally managed environments consistent with the product architecture and declared product functionality. It also includes interaction with operating systems, firmware, registries, identity services, logging systems, attestation systems, and other environmental components, even where such components remain outside the scope of the present document.

The present document addresses the intended purpose and reasonably foreseeable use of the following product categories:

- Virtualisation Execution Stack (VES); and
- Container Execution Stack (CES).

The detailed product architecture, in-scope components, and security-relevant environmental dependencies are specified in clause 4.2.

4.1.2 VES

4.1.2.1 Intended purpose and reasonably foreseeable use of the VES

The intended purpose of a VES is to enable the creation, execution, isolation, administration, and management of virtualised workloads on one or more physical hosts.

A VES may include:

- a Hypervisor, which provides workload isolation, resource partitioning, and execution control; and
- a Management and Orchestration (M&O) System, which provides provisioning, configuration, administration, scheduling, and lifecycle management functions for virtualised resources and hosts.

Reasonably foreseeable use of a VES includes:

- deployment in Type I, Type II, or Hybrid virtualisation architectures;
- administration through local or remote management interfaces;
- operation with one or more management systems coordinating one or more hypervisors;
- operation in environments where virtualised workloads share compute, memory, storage, and network resources under Hypervisor control.

4.1.2.2 Intended purpose and reasonably foreseeable use of the Hypervisor

The intended purpose of the Hypervisor is to provide the execution environment for virtual machines and to enforce isolation, execution control, and resource partitioning between virtualised workloads.

This includes, as applicable:

- controlling execution of guest virtual machines;
- mediating access to CPU, memory, I/O, storage, and interrupt-handling resources;

- enforcing isolation between guest workloads and between guest workloads and privileged Hypervisor functions;
- managing virtual-machine lifecycle operations; and
- supporting security functions native to the Hypervisor, such as logging, integrity verification, update handling, or virtualised trust services.

4.1.2.3 Intended purpose and reasonably foreseeable use of the M&O system

The intended purpose of the M&O System is to administer, configure, coordinate, and manage one or more Hypervisors and the virtualised infrastructure under their control.

This includes, as applicable:

- provisioning and configuration of virtualised resources and hosts;
- workload placement, scheduling, automation, and policy enforcement;
- lifecycle management of hosts and virtual machines;
- administration through interfaces used by operators or management agents; and
- orchestration of platform-wide functions such as updates, logging, monitoring, or trust-policy distribution.

4.1.3 CES

4.1.3.1 Intended purpose and reasonably foreseeable use of the CES

The intended purpose of a CES is to enable the execution, isolation, management, and orchestration of containerized workloads.

A CES may include:

- a Container Runtime System (CRS), which provides the execution and isolation environment for containers;
- a Container Engine (CE), which provides image handling and container lifecycle management functions; and
- a Container Orchestrator (CO), which provides deployment, scheduling, scaling, and policy enforcement across one or more nodes.

Reasonably foreseeable use of a CES includes:

- deployment on a single host or across multiple hosts or cluster nodes;
- use of local or remote administrative and control interfaces;
- operation with registries, orchestration systems, logging systems, monitoring systems, secret-management systems, or identity services;
- operation in environments where multiple containerized workloads share compute, memory, storage, and network resources.

4.1.3.2 Intended purpose and reasonably foreseeable use of the CRS

The intended purpose of the CRS is to execute containers and enforce isolation and resource control for containerized workloads on a host.

This includes, as applicable:

- creating, starting, stopping, and deleting containers;
- enforcing workload isolation and resource-control mechanisms;

- exposing runtime interfaces used by higher-level management components;
- protecting runtime configuration, control logic, and security-relevant state; and
- supporting native security functions such as logging, integrity verification, and update handling.

4.1.3.3 Intended purpose and reasonably foreseeable use of the CE

The intended purpose of the CE is to manage container images and container lifecycle operations above the CRS.

This includes, as applicable:

- building, pulling, pushing, and storing container images;
- invoking and configuring CRS operations for container creation, execution, and termination;
- managing host-level container networking and storage integration; and
- managing engine-supported plugins, extensions, or other CE-native integration functions.

4.1.3.4 Intended purpose and reasonably foreseeable use of the CO

The intended purpose of the CO is to coordinate deployment and lifecycle management of containerized workloads across one or more nodes.

This includes, as applicable:

- workload scheduling, scaling, and placement;
- admission control, policy enforcement, and cluster-level configuration management;
- management of service exposure, network policies, and workload recovery behaviour; and
- orchestration of updates, secrets handling, or cluster lifecycle functions where such functions are part of the product.

4.2 Product architecture

4.2.1 General distinction of components

The present document distinguishes between product components (within the scope of the present document) and environmental dependencies (outside the scope of the present document but potentially security-relevant):

- In-scope components represent the functional building blocks that together define the product boundary as considered in the present document. The requirements specified in the present document apply directly to these components.
- Environmental dependencies, by contrast, are not part of the product itself but belong to its operational environment (e.g. host operating system, firmware, or external services) (see clause 4.3).

4.2.2 VES

4.2.2.1 Architectural types and virtualisation techniques of VES

Depending on the deployment model and integration with the host operating environment, the VES can be implemented in one of the following architectural forms:

- Type I (Bare-metal hypervisor): a hypervisor in which the virtualisation layer runs directly on the physical hardware and supervises the software components used to create, manage, or support virtual machines.

- Type II (Hosted hypervisor): a hypervisor that executes as an application or service on top of a general-purpose host operating system and relies on the host OS for functions such as device drivers, resource management, or update handling.
- Hybrid hypervisor: a hypervisor architecture that combines characteristics of Type I and Type II architectures and may rely on a general-purpose OS, a privileged virtualised execution environment, host services, driver components, management components, or orchestration components for device drivers, management, or lifecycle functions.

The architectural forms above may use one or more virtualisation techniques, including:

- Full virtualisation: a technique where the hypervisor presents virtualised hardware interfaces so that unmodified guest operating systems can run as if on physical machines.
- Paravirtualisation: a technique where the guest operating system or a guest-resident virtualisation support component interacts with the hypervisor through defined interfaces instead of relying only on fully emulated hardware.
- Combined virtualisation models: an implementation using full virtualisation together with paravirtualised components, guest-resident virtualisation support components, or equivalent mechanisms.

4.2.2.2 In-scope components of the VES

4.2.2.2.1 Hypervisor

The following Hypervisor components are within the scope of the present document:

- Hypervisor core kernel or microkernel: central Hypervisor component responsible for enforcing core execution-control, isolation, memory-management, CPU-scheduling, interrupt-handling, and privileged resource-management functions.
- Virtual Machine Monitor (VMM) and guest VM lifecycle controller: components responsible for executing or mediating guest virtual machine execution and managing guest VM lifecycle operations, including creation, start, stop, suspension, resumption, migration where supported, and deletion.
- Resource scheduling components (CPU, memory, I/O): components responsible for allocating, scheduling, limiting, prioritizing, or accounting for access to CPU, memory, I/O, storage, interrupt-handling, or equivalent physical or virtual resources assigned to guest virtual machines.
- Device virtualisation modules (e.g. disk controllers, SR-IOV support, paravirtualised device backends, host-side drivers, or equivalent host-side components that directly interact with the Hypervisor): components that provide, mediate, or support virtualised access to devices or device functions for guest virtual machines.
- Paravirtualisation and guest-resident virtualisation support components (when applicable): components used to support paravirtualised or guest-resident virtualisation functions, where they are delivered by the Hypervisor manufacturer or are under the responsibility of the manufacturer, including:
    - Backend paravirtualisation drivers and services running in the host, in a privileged virtualised execution environment, in a management domain, in a driver domain, or in an equivalent component.
    - Frontend paravirtualisation drivers provided by the Hypervisor manufacturer for installation in guest VMs.
    - Guest-resident virtualisation support components, such as a paravisor or equivalent component, where supplied, loaded, authorized, constrained, or transitioned to by the Hypervisor or another component under the responsibility of the manufacturer.
    - Inter-VM or host-guest communication mechanisms used for paravirtual device operations (e.g. shared memory channels, ring buffers, grant tables, virtio queues).
    - A privileged VM, management domain, driver domain, general-purpose host OS component, or equivalent component only to the extent that it hosts Hypervisor-critical management or device-backend services and is part of the declared product or under the responsibility of the manufacturer.

NOTE: Guest-OS-native or third-party paravirtualisation drivers and guest-resident virtualisation support components not supplied by the Hypervisor manufacturer remain out of scope. The security boundaries and VM isolation over all host-guest, inter-VM, or equivalent communication channels are expected to be enforced by the Hypervisor or by other components identified in the declared product architecture, irrespective of the origin of the guest-side component.

- Logging and auditing subsystems native to the hypervisor: components responsible for generating, storing, protecting, or exporting security-relevant audit records and operational events produced by Hypervisor functions.

- Security update enforcement mechanisms: components or functions responsible for supporting, verifying, installing, activating, restricting, rolling back, or recording updates of Hypervisor components, where such update functions are part of the declared Hypervisor product or under the responsibility of the manufacturer.

4.2.2.2.2 Management and Orchestration System

The following Management and Orchestration (M&O) components are within the scope of the present document:

- Management and Orchestration components: This is the system itself, including the main logic for infrastructure automation and management.

- Policy enforcement, scheduling, and automation engines used for orchestration

- Hypervisor management interface or CLI/API (local or remote): This is the external interface through which administrators or the Orchestration system interact with the Hypervisor. Depending on the declared product architecture, such an interface may be part of the Hypervisor component, part of the M&O System, or shared between them in a combined VES product. In such cases, the interface is considered within the scope of the component or components for which it provides management, control, or orchestration functions. The applicable requirements and assessment provisions are determined by the component boundary declared for the product.

Deployment models of M&O system may include:

- Dedicated management host: The M&O system runs on separate physical hardware used exclusively for management and administration functions.

- Management cluster: The M&O system runs as one or more virtual machines on a dedicated cluster reserved for management and platform services.

- Virtualised M&O node: The M&O system runs as a virtual machine on the same virtualised infrastructure that it manages, placed on a host reserved for platform and management services.

Figure 4.2.2.2.2-1 and Figure 4.2.2.2.2-2 highlight in blue the components that are within the scope of the present document. Components not highlighted in blue are considered out of scope. The system boot layer, the infrastructure and VMs shown in the figures are Operational Environment components unless they are part of the declared product or under the responsibility of the manufacturer. Requirements that rely on boot verification are assessed through the corresponding product capability and documented Operational Environment dependencies, rather than by treating the entire system boot layer as in scope.

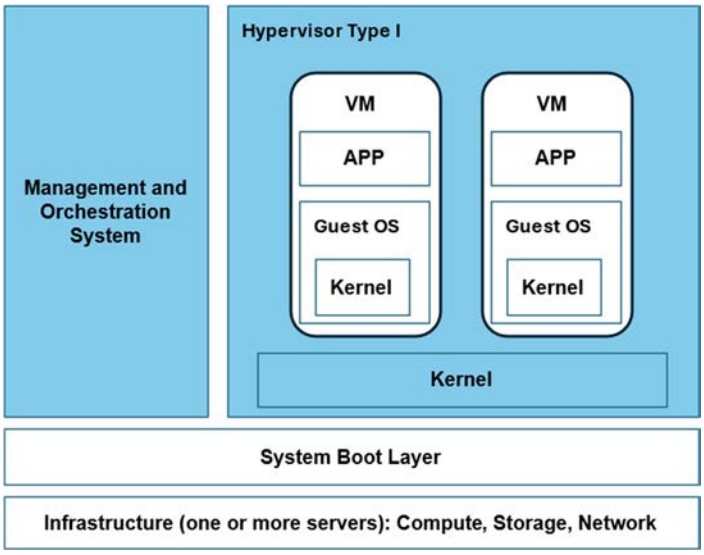

Figure 4.2.2.2.2-1: VES - Type I Hypervisor components

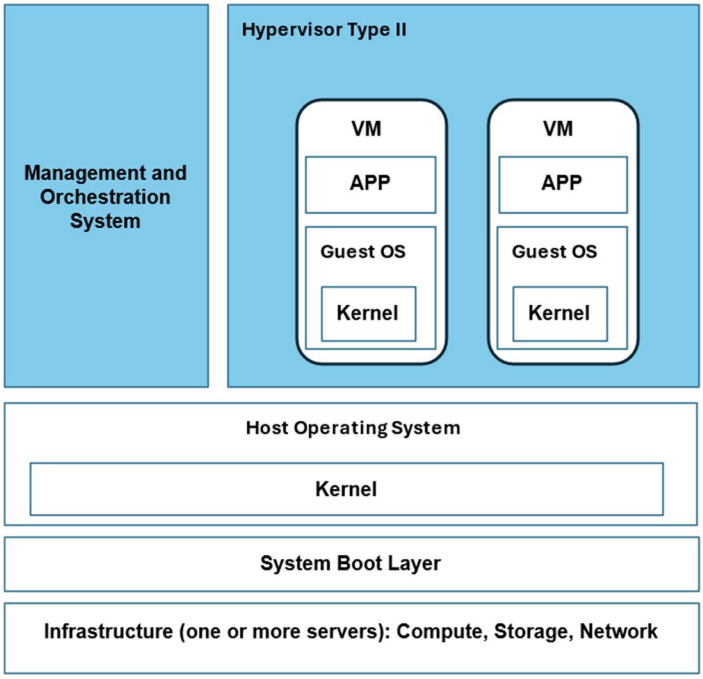

Figure 4.2.2.2.2-2: VES - Type II Hypervisor components

4.2.2.3 Security-relevant environmental dependencies (out of scope)

The following are considered outside the scope of the present document but may be security-relevant environmental components:

- The Host Firmware and BIOS/UEFI.
- The Host Operating System, when it is not bundled as part of the hypervisor.
- Guest Operating Systems and virtualised application layers, except paravirtualised drivers or guest-resident virtualisation support components that are supplied by the Hypervisor manufacturer or are under the responsibility of the manufacturer.
- Virtual network infrastructure beyond the hypervisor switch (e.g. SDN controllers).
- External logging platforms (e.g. SIEMs).

- External attestation systems.
- Hardware-level security controls (TPM, HSM) not embedded in the hypervisor or used to create a virtual equivalent (vTPM).
- Identity management systems, secrets vaults, and PKI.
- Processes, tools, environments, or development/packaging formats whose primary purpose is to develop, build, or package guest operating system artefacts, including but not limited to OS installation media, virtual disk images, VM templates, build pipelines, compilers, and related image-building utilities.
- Management and Orchestration (M&O) System provided independently by a third party.

4.2.3 CES

4.2.3.1 In-scope components

The following components are in the scope of the present document:

- CRS: it is the lowest-level layer responsible for the execution and isolation of containers:
  - The core binary (e.g. runc, crun).
  - Enforcement modules that manage container isolation, resource control, and sandboxing through OS or application-level mechanisms (e.g. namespaces, microVMs, seccomp, cgroups).
  - The API and services that manage the runtime and its containers on a host (e.g. containerd, CRI-O).
- Container Engine: it is the mid-level layer that manage the container lifecycle:
  - The engine and its interfaces for building, pulling, pushing, and running container images (e.g. Docker, Podman, containerd).
  - Management of container images including the verification of signature at the host level.
  - Management of container networking at the host level (e.g. creation of network interfaces, bridge networks, and network namespace assignment).
  - Management of volumes for persistent or shared storage.
  - Management of plugins as supported by the engine.
- Container Orchestrator: it is the top-level layer that manages the entire lifecycle of applications and services across a cluster of machines (e.g. Kubernetes®):
  - Features that ensure containers are deployed securely.
  - Image signature verification at the cluster level, secure scheduling to trusted nodes, and secrets management.
  - Policy enforcement that enforces security and isolation policies across the entire cluster (e.g. pod-level security policies and network isolation rules).
  - Lifecycle management that handles the scaling, networking, load balancing, and self-healing of containerized applications.

Where a software component, package, daemon, service, or interface provides functions corresponding to more than one CES layer, the relevant functions are considered according to the CRS, CE, or CO role they perform in the declared product architecture. In such cases, the same software package may provide evidence for more than one component role. The conformity assessment documentation identifies the relevant functions, interfaces, security boundaries, and responsibility of the manufacturer.

Figure 4.2.3.1-1 highlights in blue the components that are within the scope of the present document. Components not highlighted in blue are considered out of scope.

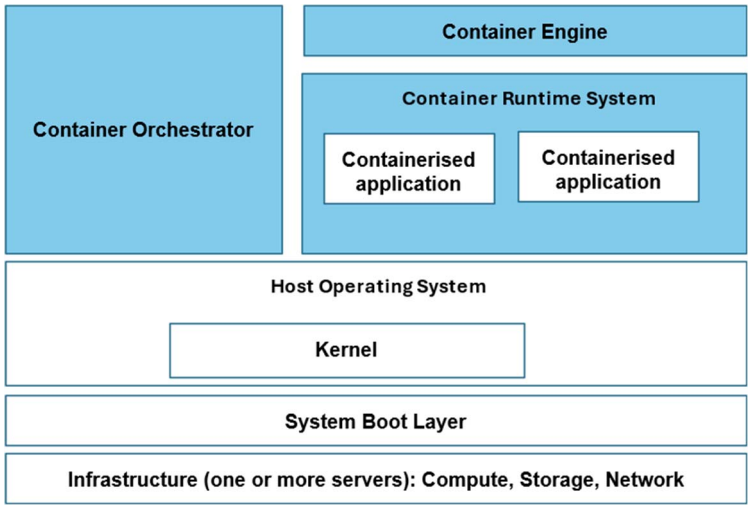

Figure 4.2.3.1-1: CES components

4.2.3.2 Security-relevant environmental dependencies (out of scope)

The following are considered outside the scope of the present document but may be security-relevant environmental components:

- The Host Operating System on which the CRS executes (OS kernel, drivers).
- Container registries (public/private).
- Container image build pipelines (CI/CD Pipelines).
- Third-party monitoring or security agents installed at runtime.
- Identity and Access Management systems used for authentication/authorization.
- Underlying hardware platform (TPM, HSM, hardware root-of-trust).
- Processes, tools, environments, or development/packaging formats whose primary purpose is to develop, build, or package container artefacts, including but not limited to build pipelines, CI/CD systems, compilers, language-specific package managers, container image builders, image layers, and image manifests.
- Container Engine (CE) and Container Orchestrator (CO) provided independently by a third party.

4.2.4 Deployment of M&O, CE, and CO outside the local execution environment

The in-scope components defined in the present document may, in some product architectures, be provided outside the same local execution environment as the rest of the product.

For a Virtualisation Execution Stack (VES), this applies to the Management and Orchestration (M&O) System.

For a Container Execution Stack (CES), this applies to the Container Engine (CE) and the Container Orchestrator (CO).

Where such M&O, CE, or CO functionality is part of the declared product functionality and is developed by the manufacturer, or under the responsibility of the manufacturer, it is considered in scope under the present document.

Where such M&O, CE, or CO functionality is provided independently by a third party and is not part of the declared product functionality under the responsibility of the manufacturer, it is considered an environmental dependency and is out of scope under the present document.

Accordingly, a conformance claim under the present document applies only to M&O, CE, or CO functionality that is part of the declared product functionality and is developed by the manufacturer, or under the responsibility of the manufacturer. No conformance claim under the present document is made for independently provided third-party services.

4.2.5 Product variants and conforming products

4.2.5.1 Definition of conforming products

The present document covers products implementing Virtualisation or container execution functionalities as described in clauses 4.2.2 and 4.2.3.

A conforming product includes at least one of the following core components:

- the Hypervisor, for products implementing a Virtualisation Execution Stack (VES); or
- the Container Runtime System (CRS), for products implementing a Container Execution Stack (CES).

The Hypervisor and CRS represent the essential execution environments that enable the isolation, management, and execution of virtualised or containerized workloads. They constitute the core mandatory components covered by the present document.

Other components described in clauses 4.2.2 and 4.2.3 such as the Management and Orchestration (M&O) System, Container Engine (CE), or Container Orchestrator (CO), are within the scope of the present document if they are part of the product.

Products that do not include at least a Hypervisor or a Container Runtime System (CRS) are not considered Virtualisation Execution Stacks or Container Execution Stacks within the meaning of the present document and therefore fall outside its scope.

4.2.5.2 Product variants

Products conforming to the present document may therefore be implemented in one or more of the following variants:

- Hypervisor only: A product implementing solely the hypervisor component.
- Hypervisor + M&O System: A product integrating both the hypervisor and its management and orchestration components.
- Container Runtime System (CRS) only: A product implementing the container execution environment without including the container engine or orchestrator.
- CRS + Container Engine (CE): A product combining the low-level runtime with the mid-level engine responsible for image handling and lifecycle management.
- CRS + Container Engine + Container Orchestrator (CO): A product integrating all three layers of the container execution stack.
- CRS + Container Orchestrator (CO): a product integrating the runtime directly with an orchestrator that manages container scheduling, scaling and policy enforcement.
- CRS + integrated CO with CE: a product integrating the runtime with an orchestrator that natively provides container-lifecycle and image-management functions.

Products that do not include at least a Hypervisor or a Container Runtime System (CRS) are not considered Virtualisation or Container Execution Stacks within the meaning of the present document and therefore fall outside its scope.

4.3 Operational environment

4.3.1 General Principles

Based on the out-of-scope elements identified for both the VES and the CES in clauses 4.2.2.3 and 4.2.3.2, the Operational Environment (OE) comprises components that are not part of the VES/CES but are essential to maintain the security of the in-scope components.

The manufacturer's responsibility under the present document is limited to implementing the security requirements applicable to the in-scope components defined in clauses 4.2.2.2 and 4.2.3.1.

However, for the VES/CES to operate securely and as intended, the OE components should function in a manner that supports and preserves the security of the product.

Where an applicable requirement relies on an OE component, capability, or service that is missing, incomplete, incompatible, or outside the responsibility of the manufacturer, the manufacturer identifies the dependency, the expected security properties, and any resulting limitation, assumption, or blocking condition in the product security documentation and secure deployment guidance.

The following security objectives for the Operational Environment specify the expected security properties of each OE component. These objectives do not make the manufacturer responsible for implementing OE components that are outside the declared product boundary or outside the responsibility of the manufacturer.

Ensuring that these objectives are met is the responsibility of the product or system integrator.

Compliance of the OE components with the CRA, for example using the relevant harmonised standards, is considered a precondition for secure operation where those OE components are required by the product security documentation.

4.3.2 Security Objectives for the Operational Environment

1. Boot Manager

- The OE should verify the integrity and authenticity of all firmware and boot components before transferring control to the hypervisor or CRS.
- The OE should enforce a chain of trust.
- The OE should block boot if signature or hash verification fails.

2. Host Operating System

- The OE should enforce strict process, memory, and I/O isolation between all processes.
- The OE should support software-assisted isolation.
- The OE should provide appropriate mechanisms to reduce kernel attack surface, such as syscall restriction, namespace or process isolation, capability bounding, mandatory access controls, or equivalent mechanisms where applicable.
- The OE should expose a secure entropy source to the CRS and Hypervisor.

3. Virtual or Physical Network Infrastructure (including SDN)

- The OE should segregate management, control, and data traffic using logical or physical separation.
- The OE should protect management channels with encryption and mutual authentication.
- The OE should enforce ACLs and rate limits to block unauthorized cross-plane traffic and DoS attempts.
- The OE should prevent MAC/IP spoofing and enforce integrity of network identities.
- The OE should provide accurate, authenticated time synchronisation for distributed components.

4. Identity and Access Management / PKI Services

- The OE should issue, rotate, and revoke credentials for administrators and services.
- The OE should validate certificates against a trusted chain and enforce revocation checks.
- The OE should store private keys in protected hardware or secure software vaults.
- The OE should record certificate lifecycle events for auditing.

5. Image Registry or Artefact Repository

- The OE should support verification of image integrity and authenticity metadata, such as digital signatures, trusted reference values, or equivalent metadata, for images where such verification is required by the configured trust policy or by the product security documentation.
- The OE should restrict registry access to authenticated and authorised users.

6. External Logging and SIEM Systems

- The OE should collect and store logs through authenticated and encrypted channels.
- The OE should preserve log integrity using signing or hashing mechanisms.
- The OE should restrict log access to authorized entities.
- The OE should correlate and alert on critical security events.
- The OE should synchronize event timestamps with a trusted time source.

7. External Attestation System

- The OE should validate attestation evidence and verify trust measurements.
- The OE should protect verifier credentials and trust anchors from compromise.
- The OE should detect replayed or stale attestation data.
- The OE should maintain logs of attestation requests and results.

8. Trust Anchors

- The OE should generate and protect cryptographic keys.
- The OE should expose secure APIs for key usage and measurement retrieval.
- The OE should provide measurement and reporting for trusted boot sequences.

9. Firewall / Intrusion Detection or Prevention System

- The OE should restrict external access to management and control interfaces.
- The OE should enforce segmentation between management, control, and data planes.
- The OE should detect and block anomalous or malicious network traffic.
- The OE should log all blocked events.

10. Storage Systems

- The OE should encrypt data at rest including VM or container images, configuration files, and logs.
- The OE should protect encryption keys in secure key management systems.
- The OE should securely erase deleted data to prevent recovery.

4.3.3 Reference to Relevant CRA Harmonised Standards

Compliance of the Operational Environment components with the CRA essential requirements may be presumed when the corresponding harmonised standards listed below are applied.

Where no harmonised standard yet exists, the system or product integrator should ensure equivalent security measures.

Table 4.3.3-1: Relevant CRA harmonised standards for OE components

OE Component	Relevant CRA Reference
Boot Manager	ETSI EN 304 623 [i.4] (CRA for Boot Manager)
Host Operating System	ETSI EN 304 626 [i.5] (CRA for Operating System)
Network Infrastructure (Virtual / Physical / SDN)	ETSI EN 304 625 [i.6] (CRA Network Interfaces)
Identity & Access Management / PKI	ETSI EN 304 624 [i.7] (CRA PKI)
External Logging / SIEM	ETSI EN 304 622 [i.8] (CRA SIEM)
Firewall / Intrusion Prevention System	ETSI EN 304 636 [i.9] (CRA Network Security Appliances)

4.4 Distribution of Security Functions

Where a VES or CES product relies on a Remote Data Processing Solution (RDPS) for the provision or support of one or more product functions, such reliance introduces a product-facing RDPS boundary between the local product side and the RDPS side.

The RDPS boundary may be relevant, for example, where a remote service supports management, orchestration, policy distribution, telemetry processing, attestation support, update coordination, cluster lifecycle management, or other VES/CES product functions.

Annex R applies to the product-facing RDPS boundary. It does not replace the requirements applicable to the product function itself or to the VES/CES components covered by the core clauses of the present document.

Where the M&O System, CE, or CO is part of the declared VES or CES product, the applicable requirements for that component are specified in the core clauses of the present document. A remote deployment, distributed implementation, or remote access model for such a component does not, by itself, make that component subject to Annex R.

Annex R is relevant where the product relies on an RDPS that is distinct from the M&O System, CE, or CO component already covered as part of the declared product, and where that RDPS provides or supports one or more RDPS-dependent product functions across a product-facing RDPS boundary.

4.5 Users

Manufacturer
The organization responsible for design, development, and delivery of the product, including the implementation of its security functions. The manufacturer provides the product documentation, security claims, and evidence required for assessment.

Product Integrator
The organization responsible for deploying, configuring, or combining the product into a larger system or operational environment. The product integrator ensures that:

- Deployment meets the manufacturer's documented security configuration requirements.
- Required platform capabilities, hardware features, and external trust anchors are present.
- The product is operated in accordance with its secure deployment guidance.

4.6 Use cases

4.6.1 Purpose

This clause defines representative use cases for VESs and CESs. These use cases describe typical deployment contexts that are used in the present document to determine the applicability of security requirements.

The use cases defined in this clause provide a structured basis for applying the requirements of the present document to common deployment scenarios.

The use cases defined in the present document are representative and cover the most common product context scenarios addressed by the present document. A product context that is not explicitly defined may be assigned to the closest representative use case, provided that:

- the assignment is duly justified;
- the assigned use case reflects an equivalent or higher security risk profile; and
- the assignment does not result in a reduction of the applicable security requirements or assessment expectations.

NOTE: Closest representative use case means the representative use case that most closely matches the product context in terms of threat exposure, risk assessment, and associated security profile.

4.6.2 Use Cases for VES

The following VES use cases define representative deployment contexts used in the present document to structure requirement applicability.

UC-V1: VES for isolated, non-critical functions

- Description: A VES deployed on a dedicated host in an isolated environment. The host hosts a very small number of static VMs dedicated to non-critical functions. In this context, non-critical functions refer to operations whose compromise would not result in harm to human safety, disruption of critical infrastructure services, or exposure of sensitive data belonging to multiple users.
- Example environments:
  - Standalone servers or industrial PCs deployed in physically isolated networks.
  - Edge devices or gateways with highly restricted links.
  - Personal computers or laptops running a local hypervisor in a non-production environment.
- Illustrative end user business applications:
  - Isolated diagnostic data collection points in a factory.
  - Environmental monitoring systems for non-critical building zones.
  - Local telemetry units for non-essential parameters in a grid.
  - Read-only public information displays in industrial facilities (e.g. current temperature, non-operational metrics).
  - Consumer devices such as personal computers or laptops running local virtual machines for hobbyist testing, educational purposes, or personal experimentation.
- Risk level: Low (see clause B.7.2).
- Main identified risks: These include malware exploiting unpatched systems in isolated industrial networks; unauthorized physical access to devices resulting in data corruption; and misconfigured hypervisors with default administrative credentials discovered in industrial telemetry units.

UC-V2: VES for general business workloads

- Description: A VES deployed within a shared multi-tenant environment (private, public, or hybrid cloud), hosting a heterogeneous mix of virtualised workloads (VMs) for various customers or independent internal units. The platform's primary goal is achieving resource efficiency and providing standard logical separation. This use case also includes deployments where the VES runs on connected edge hardware hosting multiple VMs (including legacy or specialized workloads) that require stronger isolation.
- Example environments:
  - Enterprise private cloud platforms shared across multiple departments.
  - Public Cloud multi-tenant IaaS/PaaS used for non-critical services.
  - On-premises Virtualisation clusters for internal workloads.
  - Internal development and test environments.
  - Industrial PCs or edge gateways hosting multiple VM workloads (e.g. legacy HMI + modern data gateway).
  - universal Customer Premises Equipment (uCPE) devices running virtualised network appliances (SD-WAN, firewall, router).
  - Edge servers in transportation or building systems.
- Illustrative end user business applications:
  - Corporate IT service platforms.
  - Enterprise middleware supporting internal operations (e.g. ERP, HR, finance systems).
  - Internal microservices for business applications.
  - Private banking or insurance back-end systems running on controlled virtualised infrastructures.
  - Virtualised network functions at branch/edge sites (firewall, router, SD-WAN).
- Risk level: Medium (see clause B.7.2).
- Main identified risks: These include VM escapes exploiting vulnerabilities in Virtualisation platforms; privilege escalation attacks enabling host-level access; unauthorized access to sensitive corporate data through misconfiguration or credential compromise; and exploitation of known vulnerabilities in virtualised management interfaces causing service outages or data breaches. Edge environments face higher exposure and less continuous supervision.

UC-V3: VES for critical workloads

- Description: A high-assurance VES forming the foundation of infrastructures that support critical or regulated workloads requiring strong isolation, resource separation, and continuous security assurance. Such environments may host sensitive applications, essential services, or externally sourced workloads whose compromise could lead to major operational disruption, data breaches, or safety impacts. This scenario demands maximum assurance isolation and strong separation of concerns, typically involving untrusted, heterogeneous tenants (distinct organizations or independent units). In certain deployments, these platforms may also operate at connected edge locations, where strong VM isolation is required to separate safety-relevant and non-safety workloads on the same hardware.
- Example environments:
  - Public IaaS/PaaS clouds where tenants are considered untrusted relative to one another.
  - Private clouds supporting multiple departments or projects with strong isolation needs.
  - Regulated private clouds (e.g. dedicated to a critical sector like energy).
  - High-density Virtualisation or micro-VM platforms supporting critical or externally validated workloads.

  - Connected edge systems requiring high-assurance separation between operational and support workloads.

- Illustrative end user business applications:
  - Telecom core clouds hosting critical network functions.
  - Government or defence virtualised infrastructures processing sensitive or classified data.
  - Financial services platforms hosted in public multi-tenant clouds.
  - eHealth platforms handling patient records (PHI).
  - Safety-related edge processing workloads where VM isolation protects operational integrity.
- Risk level: High (see clause B.7.2).
- Main identified risks: These include exploitation of hypervisor vulnerabilities enabling cross-workload or cross-domain compromise; hardware-level side-channel attacks (e.g. Meltdown, Spectre, Foreshadow) leaking sensitive data; denial-of-service attacks targeting shared compute, storage, or network planes; and unauthorized access through compromised management or orchestration components leading to system-wide outages or data exposure. When deployed at the edge, additional risks include semi-trusted network exposure and limited continuous monitoring.

4.6.3 Use Cases for CES

The following CES use cases define representative deployment contexts used in the present document to structure requirement applicability.

UC-C1: CES for isolated, non-critical functions

- Description: A CES deployed on a device with limited connectivity. This device typically performs non-critical functions, often in an isolated environment. These applications perform non-critical functions, meaning their compromise would not result in harm to human safety, disruption of critical infrastructure services, or exposure of sensitive data belonging to multiple users.
- Example environments:
  - Isolated test/dev environments.
  - Basic controllers for non-critical functions.
  - Consumer devices such as personal smartphones running isolated app containers for non-critical applications or laptops using local container engines (e.g. Docker Desktop, Podman) for personal projects.
- Illustrative end user business applications:
  - Educational or hobbyist containerized workloads.
  - Local telemetry or monitoring for non-critical parameters.
  - Entertainment or personal productivity apps packaged in containers.
- Risk level: Low (see clause B.7.3).
- Main identified risks: These include container escape vulnerabilities in container runtimes (e.g. runc); incidents of exposed container management daemons without TLS protection (e.g. Docker daemon) in isolated environments; and misconfigurations or unauthorized physical access leading to minor service disruption or data corruption.

UC-C2: CES for general business workloads

- Description: A CES deployed within a shared multi-tenant environment (private, public, or hybrid cloud), hosting a heterogeneous mix of containerized workloads (e.g. microservices, CI/CD pipelines). The tenants may belong to different organizations (e.g. a standard multi-client SaaS platform) or independent units within the same organization. The platform's primary goal is resource efficiency, agility, and standard logical separation. This use case also includes CES deployments at the network edge where the system runs manufacturer-defined or business-relevant workloads, may store or process sensitive data, and is reachable over external or semi-trusted networks, but where the consequence of compromise is operational or business disruption rather than critical safety failure.
- Example environments:
  - Enterprise private cloud infrastructures shared across multiple departments/teams (multi-tenancy at the organizational level).
  - Public Cloud containerized clusters used for non-critical, general business applications.
  - On-premises container platforms.
  - Internal microservices architectures.
  - Connected edge devices and gateways operating in semi-trusted networks (e.g. industrial IoT gateways, smart building controllers, retail analytics devices, agricultural monitoring and control systems).
- Illustrative end user business applications:
  - Corporate IT shared services.
  - Retail and e-commerce frontends.
  - Enterprise middleware platforms.
  - Back-office systems (HR, general accounting).
  - Local AI inference and analytics (e.g. video analytics, quality monitoring, sensor fusion) performed at the edge before forwarding to cloud services.
- Risk level: Medium (see clause B.7.3).
- Main identified risks: These include configuration and access-control weaknesses in container orchestration platforms (e.g. Kubernetes®); container breakouts through insecure host mount privileges or misconfigured namespaces; privilege escalation within containers or across pods; unauthorized lateral movement in internal networks; and exploitation of exposed container registries, management APIs, or CI/CD integration points. Edge deployments add risks from semi-trusted networks, weak monitoring, and securing local credentials/updates.

UC-C3: CES for critical workloads

- Description: A CES forming the foundation of platforms that support critical or regulated workloads with a consequence of failure classified as high impact. This platform demands high assurance isolation, typically involving dynamic, untrusted containers from heterogeneous tenants (distinct organizations or independent units). This also includes deployments at the edge where the containerized workloads perform safety-relevant, regulatory, or critical operational functions.
- Example environments:
  - Public container platforms hosting high-risk, multi-tenant applications.
  - Regulated private container platforms supporting essential services (e.g. energy, telecommunications).
  - Federated or cross-organization infrastructures hosting workloads from distinct trust domains.
  - Zero-trust environments where container-level isolation is required to be the strongest security boundary.

  - Edge deployments supporting critical services (e.g. industrial control gateways, medical data processing appliances, public infrastructure monitoring nodes).

- Illustrative end user business applications:
  - Government or telecom workloads with elevated assurance or compliance requirements.
  - Financial and e-identity services operated on high-assurance container infrastructures.
  - eHealth services handling patient records (PHI).
  - Critical data-analytics or control applications running in shared but tightly isolated environments.
  - Real-time safety, monitoring, or control workloads deployed on edge.
- Risk level: High (see clause B.7.3).
- Main identified risks: These include API server exposures in container orchestration platforms (e.g. Kubernetes®) enabling attackers to take control of clusters; container escape exploits abused in public cloud environments; supply chain attacks targeting public container images; and unauthorized data access between tenant workloads caused by misconfigured networking or weak isolation. Edge deployments add risks from semi-trusted networks, limited monitoring, and ensuring update/image integrity.

4.6.4 Use Case to Risk Mapping

Table 4.6.4-1: Use case to risk mapping

Product	Use Case	Description	Risk Level	Annex
VES	UC-V1	VES for isolated, non-critical functions	Low	B.7.2
VES	UC-V2	VES for general business workloads	Medium	B.7.2
VES	UC-V3	VES for critical workloads	High	B.7.2
CES	UC-C1	CES for isolated, non-critical functions	Low	B.7.3
CES	UC-C2	CES for general business workloads	Medium	B.7.3
CES	UC-C3	CES for critical workloads	High	B.7.3

5 Technical Requirements for products

5.1 Risk-Based requirement classes/categorization and application

5.1.1 Requirement Classes, Categorization and Security Profiles

5.1.1.1 Definition of Requirement Classes: Basic, Elevated, Advanced

To enable proportional application of security requirements under the CRA, requirements in the present document are expressed with requirement Classes: Basic, Elevated and Advanced. These levels indicate the depth or strength of the requirement.

Basic (B)

- Definition: The minimum class of requirement, providing fundamental protection.
- Characteristics:
  - Typically software-based or configuration safeguards.
  - Establishes a baseline protection expected in all products.

- Suitable for low-risk environments.

Elevated (E)

- Definition: Intermediate class of protection, addressing increased exposure.
- Characteristics:
  - Builds upon or replaces Basic with stronger mechanisms.
  - Leverages hardware-assisted features or stronger cryptographic protections.
  - Suitable for medium-risk environments.

Advanced (A)

- Definition: The highest class of protection, incorporating the strongest available safeguards.
- Characteristics:
  - Provides resilience against advanced threats.
  - Incorporates strong trust anchors and runtime protections, which may be hardware-based or software-based where demonstrably equivalent security outcomes are achieved in accordance with clause 6.1.2.
  - Suitable for high-risk environments.

5.1.1.2 Security Profile Definitions

To support risk-proportional application of security requirements under the CRA, the present document defines three Security Profiles (SP) for security requirements. The defined SPs support manufacturers in applying a risk-based approach, helping to determine applicable requirements aligned with the intended use case and proportionality principles under the CRA.

Table 5.1.1.2-1: Security profile definitions

Level	Description
SP 1	A minimum set of baseline cybersecurity requirements applicable to all products. These requirements are essential for fundamental security posture, particularly for use cases where compromise results in low risk (e.g. UC-V1, UC-C1). This level corresponds to products assessed as Low Risk according to the methodology in Annex B.
SP 2	Requirements for medium-risk environments. Depending on the requirement categories defined in clause 5.0.1.3: i) cumulative requirements include all Basic + Elevated elements; ii) exclusive requirements apply at the Elevated level only; iii) single-class requirements apply where indicated for SP 2 in the requirement applicability table. This corresponds to products assessed as Medium Risk in Annex B.
SP 3	Requirements for high-risk environments. Depending on the requirement categories defined in clause 5.0.1.3: i) cumulative requirements include Basic + Elevated + Advanced; ii) exclusive requirements apply at the Advanced level only; iii) single-class requirements apply where indicated for SP 3 in the requirement applicability table. This corresponds to products assessed as High Risk in Annex B.

5.1.1.3 Requirement Categorization

The requirements in the present document are defined using three categories: Cumulative, Exclusive, and Single-Class. Each category specifies how a requirement applies across the different classes (Basic, Elevated, Advanced). The following clauses describe these categories, providing guidance to implementers on how to apply them.

(1) Cumulative Requirements (Additive)

- Definition: Higher classes include all lower classes. Each class adds more rigor on top of the previous one.

- Rationale: This model applies when protections build in layers and depend on the foundation provided by lower classes.
- Application: If a manufacturer is assigned SP 3, then all requirements at Basic, Elevated, and Advanced marked as applicable in the requirement applicability table are applicable.
- Example - VM Isolation:
  - Basic: isolation between guest virtual machines at CPU, memory, I/O, and interrupt-handling levels.
  - Elevated: Basic + mechanisms to prevent a guest virtual machine from causing sustained denial of service to other guest virtual machines through shared or limited resource exhaustion beyond the configured policy, for example through quotas, rate limiting, prioritization, admission control, or controlled degraded behaviour.
  - Advanced: Elevated + mitigations against known side-channel attacks for workloads sharing hardware resources.

(2) Exclusive Requirements (Non-Additive)

- Definition: Only one class applies. Classes represent alternative approaches, not cumulative layers.
- Rationale: This model applies when requirements define alternative mechanisms or assurance levels, where applying the requirement at the class corresponding to the assigned SP supersedes the lower-class variant.
- Application: The manufacturer implements only the requirement at the class corresponding to the product's assigned SP, as indicated in the requirement applicability table.
- Example - Image Integrity Verification:
  - Basic: Hash verification of VM/container images before execution.
  - Elevated: Signature verification of images against a trusted public key.
  - Advanced: Signature verification with keys stored in protected trust material, which may be hardware-backed or software-based where equivalent key protection is demonstrated in accordance with clause 6.1.2.2.

(3) Single-Class Requirements

- Definition: A requirement is defined at only one class (Basic, Elevated, or Advanced). No variations exist at the other classes.
- Rationale: Some controls cannot be meaningfully scaled across classes and are best expressed at a single strength level.
- Application: A Single-Class requirement applies only to the SPs for which it is marked in the requirement applicability table. Where a Single-Class requirement is marked for more than one SP, the same requirement applies to each marked SP.

EXAMPLE:

- Advanced-only (SP 3): Remote attestation of boot-time or launch-time measurements of Hypervisor or CRS executable artefacts, where marked as an optional Advanced capability.

5.1.1.4 Guidance for Manufacturers

To identify applicable requirements:

1) Determine product use case and risk level (see clause 4.6 and Annex B).
2) Identify requirement categorization:

- If cumulative, implement all lower classes up to the assigned SP.
- If exclusive, implement only the class that matches the assigned SP.

- If single-class, implement the one requirement defined at that class.

5.1.2 Requirement Application

5.1.2.1 Approaches for determining security requirements

The present document defines an SP-based approach to identify the applicable security requirements. This approach is grounded in risk assessment and aligned with the CRA [i.1] proportionality principle.

The SP-based approach is applicable where the product aligns with one of the representative use cases defined in clause 4.6.

The representative use cases defined in clause 4.6 are representative and cover the most common product context scenarios addressed by the present document. A product context that is not explicitly defined in clause 4.6 may be assigned to the closest representative use case, provided that:

- the product context has an equivalent or higher security risk profile than the representative use case to which it is assigned;
- the assignment is justified by the manufacturer in the conformity assessment documentation; and
- the assignment does not reduce the applicable security requirements or assessment expectations under the present document.

5.1.2.2 SP-Based Requirement Application

This approach provides manufacturers with predefined use cases mapped to Security Profiles (SPs). Each SP bundles the appropriate requirements for the assessed risk level (Low, Medium, High).

The representative use cases defined in clause 4.6 are representative product contexts used to support risk-based alignment of requirements.

Step 1 - Determine product use case and associated risk level

Manufacturers begin by identifying the use case for their product (see reference cases in clause 4.6). The risk level for these use cases is pre-determined based on the application of the methodology in Annex B:

- If the product aligns with UC-V1 or UC-C1 → corresponds to a Low-risk use case.
- If the product aligns with UC-V2 or UC-C2 → corresponds to a Medium-risk use case.
- If the product aligns with UC-V3 or UC-C3 → corresponds to a High-risk use case.

Step 2 - Map use case and determined risk level to SP and apply requirements

Table 5.1.2.2-1: Security profile application by representative use case

Use Case	Risk Level	SP 1	SP 2	SP 3	Applicable requirements
UC-V1 / UC-C1	Low	x			Implement Basic requirements. For cumulative requirement families, apply the Basic requirements. For exclusive requirement families, apply the Basic requirement only. For single-class requirements, apply only those marked as applicable to SP 1 in the requirement applicability table.
UC-V2 / UC-C2	Medium		x		Implement cumulative Basic + Elevated requirements. For exclusive requirement families, apply the Elevated requirement only. For single-class requirements, apply only those marked as applicable to SP 2 in the requirement applicability table.

Use Case	Risk Level	SP 1	SP 2	SP 3	Applicable requirements
UC-V3 / UC-C3	High			x	Implement cumulative Basic + Elevated + Advanced requirements. For exclusive requirement families, apply the Advanced requirement only. For single-class requirements, apply only those marked as applicable to SP 3 in the requirement applicability table.

5.1.3 Applicability of requirements

For each component described in clauses 4.2.2.2 and 4.2.3.1, the corresponding security requirements are defined in the subsequent clauses of the present document.

The requirements in clause 5 are organised by product component and security function rather than by the order of CRA Annex I Part I essential requirements. This structure avoids duplication across VES and CES components and enables applicability to be determined according to the declared product boundary, component role, use case and Security Profile. Traceability between the requirements of the present document and the CRA essential cybersecurity requirements is provided in Annex A.

For a product implementing a Virtualisation Execution Stack (VES), the product shall include a Hypervisor. The requirements and assessment activities identified for the Hypervisor shall apply. Where the product also includes a Management and Orchestration (M&O) System, the corresponding requirements and assessment activities shall also apply.

For a product implementing a Container Execution Stack (CES), the product shall include a Container Runtime System (CRS). The requirements and assessment activities identified for the Container Runtime System (CRS) shall apply. Where the product also includes a Container Engine (CE) and/or a Container Orchestrator (CO), the corresponding requirements and assessment activities shall also apply.

Where requirement applicability depends on a use case defined in clause 4.6, the corresponding requirements and assessment activities shall apply for the applicable use case or use cases.

Where a requirement includes an applicability statement, condition, or limitation, the requirement applies only where the stated applicability condition is met for the declared product architecture, product configuration, exposed interface, implemented function, supported mechanism, or relied-upon capability. Where the condition is met, the requirement remains normative and shall be fulfilled.

Where a requirement depends on a supporting function, platform capability, Host OS capability, management-domain component, driver-domain component, M&O component, update mechanism, cryptographic mechanism, key-storage mechanism, attestation mechanism, logging mechanism, or other supporting mechanism, the requirement applies where that mechanism is part of the declared product or is under the responsibility of the manufacturer.

Where such a supporting mechanism is outside the declared product boundary or outside the responsibility of the manufacturer, it shall be treated as an Operational Environment dependency. The product security documentation shall identify the dependency, the expected security properties, and any resulting limitation or assumption relevant to conformity assessment.

Where a requirement is implemented through several components of the declared product, the requirement may be satisfied collectively by those components, provided that the responsible components, interfaces, security boundaries, and assessment evidence are identified in the product security documentation. In Hybrid, split-domain, or similar architectures, this may include privileged management domains, host services, driver domains, M&O components, or platform services, where they are part of the declared product or under the responsibility of the manufacturer and constrained by the product architecture.Hypervisor requirements in clause 5.2.1 apply to Type I, Type II, and Hybrid Hypervisor architectures according to the declared product architecture, declared product boundary, included components, exposed interfaces, implemented functions, supported mechanisms, applicable Security Profile, and any applicability statements specified under the relevant requirements.Where a requirement in clauses 5.2 or 5.3 requires, supports, or relies on cryptographic functionality, the product shall satisfy the requirements and assessment criteria for cryptographic mechanisms specified in Annex K.

Where the product uses a cryptographic mechanism that is not listed in the ECCG Agreed Cryptographic Mechanisms catalogue referenced by Annex K, but that is explicitly specified or referenced in clause K.3.2 for the relevant CES/VES product function or use case, that mechanism may be used only within the intended use and under the conditions specified in clause K.3.2. Where the product uses a cryptographic mechanism under an interoperability-based criterion specified in clause K.4.2, that mechanism may be used only for the identified interoperability or compatibility purpose and under the conditions specified in clause K.4.2.

Annex Y provides a mapping between the requirements of clauses 5.2 and 5.3 that require, support, or rely on cryptographic functionality and the corresponding ECCG ACM clause family, clause K.3.2 ACM-extended mechanism family, or clause K.4.2 interoperability-based mechanism family. Annex Y does not introduce additional requirements and does not modify the applicability, scope, or assurance expectations of clauses 5.2, 5.3, or Annex K.

Where the product relies on a remote data processing solution (RDPS), other than an M&O System, CE, or CO component already covered as part of the declared VES or CES product, for the provision or support of one or more product functions, the product shall satisfy the applicable supplementary requirements specified in Annex R for the product-facing RDPS boundary and the RDPS-dependent product function.

The requirements of Annex R do not replace the requirements applicable to the product function or to the M&O System, CE, or CO components covered by the core clauses of the present document. Where the M&O System, CE, or CO is part of the declared product, the corresponding core requirements of clause 5 remain applicable to those components, regardless of whether their implementation is local, distributed, or remotely deployed.

Where Annex R applies, the applicable Annex R requirement level is selected according to the Security Profile applicable to the RDPS-dependent product function: SP1 corresponds to the "-001" Annex R requirements, SP2 corresponds to the "-002" Annex R requirements, and SP3 corresponds to the "-003" Annex R requirements. Within each Annex R requirement family, the levels represent alternative levels of rigour and are not cumulative.

5.2 VES Security Requirements

5.2.1 Hypervisor Requirements

5.2.1.1 Isolation

5.2.1.1.1 General

This clause defines isolation requirements across three dimensions: (a) isolation between virtual machines, (b) isolation of hypervisor control/administrative planes, and (c) separation of network planes (management, guest, host).

5.2.1.1.2 VM Isolation

Classes are cumulative

Basic

REQ-H-VM-ISO-001: The Hypervisor shall enforce isolation between guest virtual machines (workloads) at CPU, memory, I/O, and interrupt-handling levels, ensuring that one guest virtual machine cannot read or modify resources of another guest VM, and cannot degrade the availability of another guest VM beyond what is permitted by the Hypervisor's configured scheduling and resource-sharing policies.

NOTE 1: Performance variability caused by normal scheduling and resource sharing (e.g. reduced CPU time under load) is acceptable when it remains within the Hypervisor's configured scheduling and resource-sharing policies. Temporary depletion of shared or limited resources is addressed under REQ-H-VM-ISO-002.

Elevated

REQ-H-VM-ISO-002: Where the Hypervisor exposes shared or limited resources to multiple guest virtual machines (e.g. entropy sources, shared device backends, shared I/O services), the Hypervisor shall provide a mechanism to prevent a guest virtual machine from causing sustained denial of service to other guest virtual machines through resource exhaustion beyond the configured policy, for example through quotas, rate limiting, prioritization, admission control, or controlled degraded behaviour.

Advanced

REQ-H-VM-ISO-003: The Hypervisor shall strengthen isolation with mitigations against known side-channel attacks to ensure robust separation between workloads sharing hardware resources.

Applicability of REQ-H-VM-ISO-003:

This requirement applies where the declared Hypervisor product exposes shared execution resources, shared microarchitectural resources, shared I/O resources, or scheduling and resource-sharing mechanisms for which side-channel mitigations are required by the applicable use case or Security Profile. Where effective mitigation depends on Host OS or platform mechanisms outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

REQ-H-VM-ISO-004 (Guest–Host Isolation): The Hypervisor shall enforce that all access by privileged host-environment components to guest memory, guest CPU state, or other security-relevant guest execution state is mediated through defined, auditable guest-host interfaces, and that access outside those interfaces is prevented or detected by the Hypervisor's isolation mechanisms.

Applicability of REQ-H-VM-ISO-004:

This requirement applies where the declared Hypervisor product includes, controls, or relies on privileged host-environment components, management-domain components, driver-domain components, host services, paravirtualised components, or equivalent mechanisms that can access guest memory, guest CPU state, or other security-relevant guest execution state. The requirement applies to the mechanisms by which such access is mediated, prevented, detected, authorized, constrained, or made auditable by the Hypervisor product.

Where such components or mechanisms are outside the declared product boundary or outside the responsibility of the manufacturer, they shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

NOTE 2: Where a Hypervisor loads or authorizes a guest-resident virtualisation support component, such as a paravisor or equivalent component, the defined and auditable interface obligation applies to the mechanism by which that component is loaded, authorized, constrained, or transitioned to by the Hypervisor. Direct accesses performed by that component within the guest execution environment are not considered privileged host-environment access under this requirement, provided that the component's role and boundary are identified in the product security documentation.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-VM-ISO-001	x	x	x
REQ-H-VM-ISO-002		x	x
REQ-H-VM-ISO-003			x
REQ-H-VM-ISO-004			x

5.2.1.1.3 Control Plane Isolation

Classes are cumulative

Basic

REQ-H-CP-ISO-001: The Hypervisor shall support configurations that enforce access-control mechanisms for its administrative functions and interfaces. These configurations shall enable guest virtual machines to be prevented from directly accessing or invoking administrative functions and interfaces, and shall ensure that only authenticated and authorized entities can perform management actions.

Elevated

REQ-H-CP-ISO-002: Where the declared Hypervisor product exposes network-accessible administrative or control-plane interfaces, the Hypervisor shall support configurations that enforce the use of cryptographically protected communication for those interfaces, providing confidentiality and integrity protection. The Hypervisor shall also support the use of distinct communication channels, ports, or virtual networks to enable isolation of administrative traffic from guest workloads and guest or data-plane traffic.

Applicability of REQ-H-CP-ISO-002:

This requirement applies where the declared Hypervisor product exposes network-accessible administrative or control-plane interfaces. Where separation of administrative or control-plane traffic from guest, workload, host, storage, or data-plane traffic depends on Host OS, platform, or external networking mechanisms outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

Advanced

REQ-H-CP-ISO-003: Where the declared Hypervisor product exposes network-accessible administrative or control-plane interfaces, the Hypervisor shall support configurations that enforce cryptographically protected and mutually authenticated communication channels for those interfaces, in order to reduce the risk of unauthorized access or lateral movement across tenants or system domains.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-CP-ISO-001	x	x	x
REQ-H-CP-ISO-002		x	x
REQ-H-CP-ISO-003			x

5.2.1.1.4 Network Plane Separation

Classes are cumulative

Applicability of REQ-H-NP-ISO-001 to REQ-H-NP-ISO-003:

These requirements apply where the declared Hypervisor product provides, configures, or controls virtual networking, network-plane separation, or equivalent network enforcement mechanisms for guest, host, or management traffic. Where network-plane separation is provided by mechanisms outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

Basic

REQ-H-NP-ISO-001: The Hypervisor shall maintain logical segregation between the management, guest, and host network planes. Virtual switching, routing, or equivalent network mechanisms shall ensure that traffic from one plane cannot traverse or address another plane unless explicitly permitted by configuration. These mechanisms shall enforce explicit configuration of separate network contexts for each plane.

Elevated

REQ-H-NP-ISO-002: The Hypervisor shall support configuration of explicit traffic filtering and access-control rules between network planes to prevent unauthorized or unintended cross-plane communication. These rules shall include filtering of packets at the Hypervisor's virtual network layer or at other enforcement points under the Hypervisor's control, ensuring that only management or orchestration traffic explicitly permitted by the configured policy is allowed to cross plane boundaries.

Advanced

REQ-H-NP-ISO-003: The Hypervisor shall support isolation of management traffic using dedicated physical interfaces, where such interfaces are available as part of the platform, or cryptographically protected network channels, preventing interception or modification by guest or host workloads. Where dedicated physical interfaces are not used, the cryptographic protection shall provide confidentiality, integrity, and authentication equivalent to a dedicated management network.

NOTE 1: Cryptographically protected network channels may be implemented over logically separated networks (e.g. VLAN, VXLAN, or equivalent), provided that confidentiality, integrity, and authentication are ensured by the cryptographic protection mechanism.

NOTE 2: This requirement does not imply that the Hypervisor itself includes hardware components or cannot be implemented purely in software. It applies to the Hypervisor's capability to use dedicated physical interfaces where such interfaces are available as part of the platform. Where such interfaces are not available, the requirement is addressed through support for cryptographically protected network channels.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-NP-ISO-001	x	x	x
REQ-H-NP-ISO-002		x	x
REQ-H-NP-ISO-003			x

5.2.1.2 Integrity Protection

5.2.1.2.1 Boot integrity verification

Single Class

Applicability of REQ-H-B-INT-002:

This requirement applies to Hypervisor executable artefacts delivered as part of the declared Hypervisor product.

It does not require the Hypervisor to verify firmware, platform bootloaders, Host OS components, hardware roots of trust, or other Operational Environment components, unless those components are part of the declared Hypervisor product.

Where verification is performed by a platform-provided boot mechanism or other Operational Environment mechanism, that mechanism shall be treated as an Operational Environment dependency. The product security documentation shall identify the expected verification mechanism, the expected security properties, and any assumptions required for the Hypervisor to be securely loaded.

Elevated

REQ-H-B-INT-002: The Hypervisor shall support deployment with a boot mechanism that verifies the integrity and authenticity of its executable artefacts before those artefacts are used to initialize guest workloads or expose management interfaces.

The Hypervisor shall be delivered with, or support the use of, verification or measurement metadata for those artefacts, such as digital signatures, certificates, cryptographic digests, measurements, trusted reference values, or equivalent metadata.

Where the Hypervisor includes the mechanism that loads or activates its executable artefacts, this mechanism shall prevent loading or activation of those artefacts when verification fails.

Where verification status from a platform-provided boot mechanism is made available to the Hypervisor, the Hypervisor shall use that status before initializing guest workloads or exposing management interfaces.

Where verification status from a platform-provided boot mechanism is made available to the Hypervisor, if the required verification status cannot be established or is reported as failed, the Hypervisor shall prevent initialization of guest workloads and activation of management interfaces where such enforcement is under the control of the Hypervisor.

NOTE 1: Hypervisor executable artefacts may include, depending on the Hypervisor architecture, the Hypervisor image, kernel, microkernel, virtual machine monitor, driver-domain components, or equivalent executable artefacts delivered as part of the Hypervisor.

NOTE 2: Boot mechanisms may include software-based platform mechanisms, hardware-based platform mechanisms, or a combination of both. This requirement does not require the Hypervisor to provide those mechanisms unless they are part of the declared Hypervisor product.

NOTE 3: Preventing the activation of management interfaces only concerns management interfaces for guest workloads. Other management interfaces such as update mechanisms are not considered to be in scope for prevention.

NOTE 4: Requiring verification status may be a configuration option for the administrator.

NOTE 5: A verification status of "verification disabled" from the platform-provided boot mechanism does neither count as "can not be established" nor "failed".

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-B-INT-002		x	x

5.2.1.2.2 Guest VM image integrity verification

Classes are exclusive

Basic

REQ-H-IMG-INT-001: The Hypervisor shall implement an integrity verification mechanism for guest VM images based on a trusted reference value.

The Hypervisor shall perform, or cause to be performed, this integrity verification automatically before executing or provisioning any guest VM. If the integrity verification of a guest VM image fails, the Hypervisor shall prevent execution or instantiation of that guest VM.

Elevated

REQ-H-IMG-INT-002: The Hypervisor shall perform, or cause to be performed, verification of both the integrity and authenticity of guest VM images before execution or provisioning. This verification shall establish that the image presented for execution or provisioning originates from a trusted source and has not been altered since it was approved for deployment.

The Hypervisor shall prevent execution or instantiation of any guest VM image whose integrity or authenticity cannot be successfully verified.

NOTE 1: Authenticity verification may be implemented using public-key digital signatures validated through certificate chains rooted in trusted authorities, or other cryptographic schemes that provide a verifiable binding between the guest VM image and a trusted source and that prevent undetected modification of the image, provided that equivalent assurance is achieved.

NOTE 2: An authenticated and authorized administrator may add or remove trusted authorities.

Advanced

REQ-H-IMG-INT-003: The Hypervisor shall use integrity and authenticity verification for all guest VM images before execution or provisioning, based on trust anchors stored as protected trust material. The protected trust material shall not be modifiable during normal operational state.

If verification of the integrity or authenticity of a guest VM image using the protected trust material fails, or if the verification status of the image cannot be established, the Hypervisor shall prevent execution or instantiation of that guest VM.

Where protection of trust anchors, keys, certificates, manifests, signatures, or equivalent trust material is provided by Host OS, platform, keystore, or equivalent mechanisms outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

NOTE 3: Protected trust material refers to keys, certificates or other trust anchors that are stored and managed so that they cannot be altered through unauthorized administrative or guest-facing interfaces and can only be changed by authenticated and authorized administrators with appropriate privileges. This also applies to product updates changing protected trust material. Such changes are recorded in security-relevant logs sufficient to support security auditing.

NOTE 4: These requirements apply to the guest VM image, boot artefact, or provisioning artefact as presented for execution, provisioning, redeployment, or re-instantiation. They do not require normal mutable guest storage or guest OS state to remain immutable after deployment, unless the product claims such an immutable workload model.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-IMG-INT-001	x		
REQ-H-IMG-INT-002		x	
REQ-H-IMG-INT-003			x

5.2.1.2.3 Runtime integrity protection

Single Class

Applicability of REQ-H-RP-INT-002:

Runtime integrity protection may rely on mechanisms provided by the Hypervisor, other components of the declared product, or the Operational Environment. Operational Environment dependencies shall be identified in the product security documentation.

Elevated

REQ-H-RP-INT-002: The Hypervisor shall support integrity protection for its security-critical runtime code, configuration state, and control logic used to enforce security policies or isolation.

The Hypervisor shall support mechanisms ensuring that security-critical configuration state and control logic can be modified only through authenticated and authorized interfaces or processes

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-RP-INT-002		x	x

5.2.1.2.4 Remote attestation

Single Class

Applicability of REQ-H-RA-INT-003 and REQ-H-RA-INT-004:

REQ-H-RA-INT-003 and REQ-H-RA-INT-004 are optional Advanced-class requirements. They apply only when the declared product configuration includes remote attestation as an optional capability for UC-V3 / SP3.

Advanced

REQ-H-RA-INT-003: Where remote attestation is included in the declared product configuration, the Hypervisor shall provide generation of attestation evidence describing boot-time or launch-time measurements of its security-critical executable artefacts.

This attestation evidence shall be cryptographically verifiable by an authorized verifier and bound to the Hypervisor instance for which the measurements were produced.

The Hypervisor shall protect attestation keys and measurement data against unauthorized access, disclosure, or replay.

The attestation evidence shall be limited to the minimum information necessary to verify the integrity of the measured Hypervisor executable artefacts.

Where attestation evidence relies on platform, Host OS, firmware, hardware root of trust, or external attestation services outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

Where the declared product additionally supports generation of attestation evidence describing security-critical configuration state - such as enabled security features, active isolation policies, or loaded security-relevant modules - the product may include such configuration-state measurements in the attestation evidence. Where configuration-state attestation is claimed in the product security documentation, such measurements shall be subject to the same cryptographic verifiability and key protection obligations as the boot-time measurements required by this requirement.

NOTE 1: This requirement covers the Hypervisor capability to collect, use, bind, or expose boot-time or launch-time measurements as attestation evidence. The design and operation of local or remote verifiers, attestation protocols, and relying parties is outside the scope of the present document.

NOTE 2: Hypervisor executable artefacts may include, depending on the Hypervisor architecture, the Hypervisor image, kernel, microkernel, virtual machine monitor, driver-domain components, or equivalent executable artefacts delivered as part of the Hypervisor.

NOTE 3: This requirement does not require attestation of arbitrary Hypervisor memory, guest memory, guest device state, transient scheduling state, workload-dependent runtime state, or other dynamic runtime data that is not security-critical configuration state as described in this requirement.

REQ-H-RA-INT-004: Where attestation evidence produced by the Hypervisor is intended to be used in remote verification scenarios, the Hypervisor shall support configurable privacy-preserving options, as appropriate to the attestation model, such as limiting attested fields to the minimum necessary for integrity verification, using pseudonymous or rotating attestation identifiers, separating platform identity information from integrity measurement information, configuring attestation trigger conditions and frequency, and restricting attestation access to authorized verifiers.

NOTE 4: This requirement does not mandate a specific attestation protocol or identity system.

NOTE 5: Remote attestation is optional in the Advanced class. A product may conform to Advanced without including remote attestation in the declared product configuration. Where remote attestation is not included in the declared product configuration, this limitation is identified in the product security documentation.

Requirement applicability by use case and security profile

NOTE 6: In the following table, "o" indicates an optional capability within UC-V3 / SP3. Absence of support does not preclude Advanced-class conformance.

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-RA-INT-003			o
REQ-H-RA-INT-004			o

5.2.1.3 Authentication

5.2.1.3.1 General

This clause distinguishes between authentication of administrative users and authentication of external services that access Hypervisor management functions or data.

5.2.1.3.2 Administrative Authentication

Classes are cumulative

Basic

REQ-H-ADMIN-AUTH-001: Where the Hypervisor exposes administrative interfaces (including local command-line interfaces, remote application programming interfaces, and virtual machine management consoles), the Hypervisor shall require authentication for access to those interfaces.

The Hypervisor shall enforce the use of unique, non-default credentials for administrative accounts. Where password-based authentication is supported, the Hypervisor shall enforce password complexity requirements for administrative authentication credentials and a mechanism limiting repeated failed authentication attempts, with configurable parameters.

Elevated

REQ-H-ADMIN-AUTH-002: Where the Hypervisor exposes network-accessible administrative interfaces, the Hypervisor shall support cryptographic key-based authentication for administrative access and shall allow password-based login to be disabled for those interfaces.

- Algorithms and key lengths shall conform to current cryptographic best practices, aligned with Annex K.
- Private keys used for administrative authentication shall be stored and managed in a manner that prevents unauthorized extraction or reuse by guest workloads or other unprivileged components.

Advanced

REQ-H-ADMIN-AUTH-003: Where the Hypervisor exposes administrative interfaces, the Hypervisor shall require strong authentication (as defined in clause 3.1) for all administrative interfaces.

Where certificate-based administrative authentication is used, certificate validity, expiration, and revocation status shall be verified before administrative access is granted.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-ADMIN-AUTH-001	x	x	x
REQ-H-ADMIN-AUTH-002		x	x
REQ-H-ADMIN-AUTH-003			x

5.2.1.3.3 Service Authentication

Classes are cumulative

Basic

REQ-H-SERV-AUTH-001: Where the Hypervisor exposes interfaces for external services to access management functions or data, the Hypervisor shall require authentication for all external services, including orchestration systems, monitoring tools, and backup agents, before granting access to management functions or data.

The Hypervisor shall enforce the use of distinct credentials for each service account and shall not allow built-in shared or hard-coded service credentials.

The Hypervisor shall implement an account lockout mechanism, rate limiting, backoff, throttling, or an equivalent mechanism that limits repeated failed authentication attempts for service accounts after a configurable number of consecutive failed authentication attempts. Where an account lockout mechanism is used, the Hypervisor shall support either administrative unlock or automatic unlock after a configurable delay.

Elevated

REQ-H-SERV-AUTH-002: Where the Hypervisor exposes interfaces for external services to access management functions or data, the Hypervisor shall require authenticated access for all external services and shall support certificate-based mutual authentication for services accessing management functions or data.

Certificates used for service authentication shall be verified for validity and expiration before access is granted.

- Algorithms and key lengths shall conform to current cryptographic best practices, aligned with Annex K.
- Private keys used for service authentication shall be stored and managed in a manner that prevents unauthorized extraction or reuse by guest workloads or other unprivileged components.

Advanced

REQ-H-SERV-AUTH-003: Where the Hypervisor exposes interfaces for external services to access management functions or data, the Hypervisor shall enforce certificate-based mutual authentication for all service interactions that access management functions or data, with certificate paths validated against a trusted public key infrastructure.

Certificates used for service authentication shall be bound to specific service identities and shall be verified for validity, expiration, and revocation status before access is granted.

- Algorithms and key lengths shall conform to current cryptographic best practices, aligned with Annex K.
- Private keys used for service authentication shall be stored and managed in a manner that prevents unauthorized extraction or reuse by guest workloads or other unprivileged components.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-SERV-AUTH-001	x	x	x
REQ-H-SERV-AUTH-002		x	x
REQ-H-SERV-AUTH-003			x

5.2.1.4 Authorization

5.2.1.4.1 General

This clause distinguishes between authorization of administrative users and authorization of external services that access Hypervisor management functions or data.

NOTE: In this clause, granular authorization means that permitted actions are bound to authenticated identities and least-privilege permissions by default. Fine-grained, policy-driven authorization means that decisions can use more detailed policy rules, separation-of-duties constraints, contextual attributes, or integration with external policy decision systems where supported by the declared product configuration.

5.2.1.4.2 Administrative Authorization

Classes are cumulative

Basic

REQ-H-ADMIN-AUTHZ-001: Where the Hypervisor exposes administrative interfaces, the Hypervisor shall restrict all administrative actions to authenticated accounts and enforce a default-deny policy for access to management functions.

Elevated

REQ-H-ADMIN-AUTHZ-002: Where the Hypervisor exposes administrative interfaces, the Hypervisor shall enforce granular authorization for administrative accounts, binding permitted actions to the authenticated identity and enforcing least-privilege permissions by default.

Advanced

REQ-H-ADMIN-AUTHZ-003: Where the Hypervisor exposes administrative interfaces, the Hypervisor shall enforce fine-grained, policy-driven authorization for administrative accounts, including support for separation of duties and context-aware enforcement where applicable.

Where time-bound or just-in-time privilege elevation is used, the Hypervisor shall support enforcement of time-limited elevation and revocation, either through native capabilities or through integration with external identity, access-management, or policy decision systems.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-ADMIN-AUTHZ-001	x	x	x
REQ-H-ADMIN-AUTHZ-002		x	x
REQ-H-ADMIN-AUTHZ-003			x

5.2.1.4.3 Service Authorization

Classes are cumulative

Basic

REQ-H-SERV-AUTHZ-001: Where the Hypervisor exposes interfaces for external services to access management functions or data, the Hypervisor shall enforce a default-deny policy for all service accounts, ensuring that service accounts are only permitted to perform actions explicitly authorized.

Elevated

REQ-H-SERV-AUTHZ-002: Where the Hypervisor exposes interfaces for external services to access management functions or data, the Hypervisor shall enforce granular authorization for service accounts, binding allowed actions to their authenticated identity. Each service account shall be configured with least-privilege permissions by default.

Advanced

REQ-H-SERV-AUTHZ-003: Where the Hypervisor exposes interfaces for external services to access management functions or data, the Hypervisor shall enforce fine-grained, policy-driven authorization for service accounts, binding permitted actions to authenticated service identities and supporting context-aware enforcement where applicable.

The Hypervisor shall support integration with external authorization systems where such integration is part of the declared product configuration.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-SERV-AUTHZ-001	x	x	x
REQ-H-SERV-AUTHZ-002		x	x
REQ-H-SERV-AUTHZ-003			x

5.2.1.5 Confidentiality Protection

Classes are cumulative

Basic

REQ-H-CONF-001 (Access control for sensitive data): The Hypervisor shall provide access-control mechanisms to restrict access, through Hypervisor functions and interfaces, to sensitive data associated with the Hypervisor product, including guest VM images, snapshot data, configuration files, credentials, and security-relevant logs. Where such data is stored, managed, or exposed as part of the product, access shall be permitted only to authenticated and authorized entities through controlled interfaces.

NOTE 1: This requirement applies to access through Hypervisor functions and interfaces to sensitive data that is stored, managed, or exposed as part of the product. It does not apply to copies of such data held exclusively by out-of-scope environmental components, such as a host operating system or external storage or network services.

Elevated

REQ-H-CONF-002 (Data at rest): The Hypervisor shall support configurations that enforce encryption of sensitive data at rest for data that it stores or manages, including VM images, snapshot data, configuration files, credentials, and security-relevant logs.

Encryption keys used for data-at-rest protection shall be stored and managed in a manner that prevents access by guest workloads and other unprivileged components.

Applicability of REQ-H-CONF-002:

This requirement applies where the declared Hypervisor product stores, manages, or exposes sensitive data at rest, including VM images, snapshot data, configuration files, credentials, security-relevant logs, or equivalent Hypervisor-managed data. Where storage encryption, key protection, or key management is provided by storage, Host OS, platform, external key-management service, or other mechanisms outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

NOTE 2: In multi-tenant deployments, protection of data-at-rest keys may be strengthened through tenant-specific key-management models, such as Bring Your Own Key (BYOK), Hold Your Own Key (HYOK), or equivalent approaches providing comparable tenant separation and key-control assurances.

REQ-H-CONF-003 (Data in transit): The Hypervisor shall support the use of encrypted communication for management interfaces, for VM migration traffic, and for inter-VM traffic where such traffic is mediated by Hypervisor-controlled virtual networking components.

When these communications are configured to use encrypted transport, the Hypervisor shall prevent fallback to unencrypted or weakly protected channels.

NOTE 3: This requirement does not require the Hypervisor to encrypt all guest-originated application traffic traversing a virtual switch. Where traffic content and protocol selection are controlled by the guest workload, the decision to encrypt that application traffic remains under the control of the guest or its operator. The requirement applies to Hypervisor management traffic, VM migration traffic, and Hypervisor-provided or Hypervisor-configured protected communication channels.

Advanced

REQ-H-CONF-004 (Data in use - confidential execution): The Hypervisor shall support provisioning and management of execution environments for guest workloads in which confidential data associated with those workloads is protected, during execution, from observation, inspection, or access through the Hypervisor, the host operating system, and administrative interfaces.

On platforms that provide mechanisms for confidential execution, the Hypervisor shall be able to use these mechanisms to enforce the confidentiality of guest memory and associated processor state.

Where confidential execution depends on platform mechanisms, hardware-assisted isolation, firmware, Host OS services, or external trust services outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-CONF-001	x	x	x
REQ-H-CONF-002		x	x
REQ-H-CONF-003		x	x
REQ-H-CONF-004			x

5.2.1.6 Availability and Resilience

Classes are cumulative

Basic

REQ-H-AVAIL-001: The Hypervisor shall detect failures of internal management and control-plane components and shall support recovery of these components at the component level, without initiating a full host reboot, except where the nature of the failure precludes component-level recovery without loss of system integrity. The Hypervisor shall maintain the operational state of running guest virtual machines during and after any component-level recovery process.

NOTE 1: Component-level recovery may encompass mechanisms such as watchdog-triggered restart of management daemons, hot-restart of control-plane services, or in-process fault isolation. Failure modes for which the product design permits only full host reboot as the safe recovery path may be identified in the technical documentation accompanying the product.

Elevated

REQ-H-AVAIL-002: The Hypervisor shall provide an automated recovery mechanism for guest virtual machines affected by host or Hypervisor service failure, such that affected guest virtual machines are automatically returned to an operational state.

NOTE 2: This may be achieved by automatic restart initiated by the Hypervisor, or by integration with an external supervisor/lifecycle manager that performs the restart or recovery action using Hypervisor-provided interfaces.

Advanced

REQ-H-AVAIL-003: For selected guest virtual machines, the Hypervisor shall support a continuity mechanism that reduces service interruption compared to a full restart.

Where continuity depends on Host OS, storage, clustering, platform, or external lifecycle-management mechanisms outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

NOTE 3: Continuity mechanisms may include checkpoint/restore, warm standby, live migration, or runtime-state replication where supported by the product context.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-AVAIL-001	x	x	x
REQ-H-AVAIL-002		x	x
REQ-H-AVAIL-003			x

5.2.1.7 Logging

Classes are cumulative

Basic

REQ-H-LOG-001: The Hypervisor shall generate time-stamped audit logs for security-relevant administrative actions and security events, including at least authentication attempts, configuration changes, and virtual machine lifecycle operations.

Each audit record shall include, as applicable to the event type, an event type identifier, the time of the event, the outcome of the event, and the identity of the associated user, service, or process.

NOTE: The set of administrative actions and security events that are considered security-relevant may vary depending on the Hypervisor implementation and its exposed management interfaces. The product security documentation should identify which actions/events are treated as security-relevant for audit logging and explain any product-specific additions beyond the minimum set listed above.

Elevated

REQ-H-LOG-002: The Hypervisor shall protect audit logs from unauthorized access, modification, or deletion. Access to audit logs shall be restricted to authorized administrative roles or processes.

The Hypervisor shall support secure export of audit logs to external log management or security information and event management systems using authenticated and encrypted communication channels.

Where the Hypervisor exposes tenant-visible logs or log-derived information in multi-tenant environments, it shall maintain separation between operator audit logs and tenant-visible logs or telemetry.

Advanced

REQ-H-LOG-003: The Hypervisor shall support mechanisms to protect audit logs generated under REQ-H-LOG-001 against undetected modification, deletion, or reordering. The protection shall enable verification of the integrity and time or sequence of security-relevant audit records, using cryptographic protection, trusted time-stamping, append-only or tamper-evident storage, secure export to an external log management system, or an equivalent mechanism.

Where the protection relies on mechanisms outside the declared product boundary or outside the responsibility of the manufacturer, the mechanism shall be treated as an Operational Environment dependency. The product security documentation shall identify the dependency and the expected security properties.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-LOG-001	x	x	x
REQ-H-LOG-002		x	x
REQ-H-LOG-003			x

5.2.1.8 Secure Update

Classes are cumulative

Applicability of REQ-H-UPD-001 to REQ-H-UPD-003:

These requirements apply to updates of Hypervisor components delivered as part of the declared Hypervisor product.

Where update delivery, verification, installation, rollback protection, or live patching is performed by a Host OS update service, package manager, platform update service, appliance update mechanism, M&O system, orchestration-based update pipeline, or other Operational Environment mechanism outside the declared Hypervisor product or outside the responsibility of the manufacturer, that mechanism shall be treated as an Operational Environment dependency. The product security documentation shall identify the expected update mechanism, the expected security properties, and any assumptions required for secure update of the Hypervisor.

Where update functions are provided collectively by the Hypervisor and other components under the responsibility of the manufacturer, the product security documentation shall identify the responsible components, interfaces, security boundaries, and assessment evidence.

Basic

REQ-H-UPD-001: The Hypervisor shall be delivered in a form that supports authenticity and integrity verification of Hypervisor updates before installation or activation, using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.

Where the Hypervisor includes the mechanism that verifies, installs, or activates its updates, this mechanism shall prevent installation or activation of an update when authenticity or integrity verification fails and shall record a security-relevant event.

Elevated

REQ-H-UPD-002: The Hypervisor shall support rollback protection for Hypervisor updates, where rollback protection is provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer.

Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed Hypervisor updates shall be prevented.

Authorized rollback to a previously trusted Hypervisor version shall only be permitted when the rollback image or package is verified for integrity and authenticity, the rollback action is explicitly authorized by an administrator with appropriate privileges or an equivalent authorized control function, and the rollback operation is recorded as a security-relevant event.

Advanced

REQ-H-UPD-003: The Hypervisor shall provide configuration or administrative controls allowing the operator to enable, disable, or restrict live patching according to the operational security policy.

NOTE: Live patching may introduce security and operational trade-offs. This requirement does not imply that live patching is preferable to reboot-based update. The decision to enable live patching is an operational policy decision based on the product context, availability needs, and risk assessment.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-UPD-001	x	x	x
REQ-H-UPD-002		x	x
REQ-H-UPD-003			x

5.2.1.9 Secure Configuration and Default

Classes are cumulative

Basic

REQ-H-CFG-001: The Hypervisor shall, by default, disable remote administrative access pathways and insecure management interfaces that are not required for initial provisioning of the Hypervisor product. This includes remote shell access without strong authentication, legacy or unencrypted management APIs, and vendor-specific diagnostic ports that are not required for secure operation.

The Hypervisor shall provide configuration mechanisms that allow administrators to explicitly enable only those interfaces and services that are necessary for the intended Virtualisation, orchestration, and core host management operations.

Elevated

REQ-H-CFG-002: The Hypervisor shall validate configuration parameters before applying them and shall implement configuration validation logic that detects and prevents configurations which would disable or bypass security mechanisms or violate guarantees for virtual machine isolation, control-plane protection, or minimum resource allocations for security and management functions.

When a configuration change is rejected due to a security-related validation failure, the Hypervisor shall provide clear feedback to the administrator indicating the reason for rejection.

Advanced

REQ-H-CFG-003: The Hypervisor shall support the definition and use of security configuration baselines that specify secure settings for critical Hypervisor components, including at least I/O memory management unit configuration, virtual switch policies, and host firewall rules, as implemented within the Hypervisor product.

The Hypervisor shall provide capabilities to automatically validate current configuration settings against a selected security baseline and shall generate alerts or audit events when non-compliant settings are detected.

Applicability of REQ-H-CFG-003:

This requirement applies to security configuration baselines for critical Hypervisor components and security-relevant settings that are part of the declared Hypervisor product or configurable through Hypervisor-controlled interfaces. Where baseline settings refer to Host OS, platform, network, firewall, or hardware mechanisms outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-CFG-001	x	x	x
REQ-H-CFG-002		x	x
REQ-H-CFG-003			x

5.2.1.10 Data Minimisation

Single Class

Basic

REQ-H-DM-001: The Hypervisor shall limit log content, telemetry, and diagnostic data to information necessary for operation, troubleshooting, and security monitoring.

Logs, telemetry, and diagnostic outputs produced by the Hypervisor shall not include plaintext authentication credentials, full cryptographic keys, guest workload payload data, or virtual machine memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data, unless such inclusion is explicitly enabled for a specific debugging purpose by an administrator.

Payload or application data originating from guest workloads shall not be included in Hypervisor logs, telemetry, or diagnostic outputs by default. Where the Hypervisor supports inclusion of such payload content for debugging purposes, this behaviour shall only be enabled through explicit administrative configuration and shall be clearly indicated to the administrator.

REQ-H-DM-002: The Hypervisor shall provide administrative controls to restrict and, where necessary, disable the collection of specific categories of log, telemetry, and diagnostic data, including the ability to disable collection of particular log types, telemetry streams, metrics, or debug payload capture.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-DM-001	x	x	x
REQ-H-DM-002	x	x	x

5.2.1.11 Decommissioning

Single Class

Basic

REQ-H-DECOM-001: The Hypervisor shall provide a mechanism to decommission a guest virtual machine that renders the associated images, snapshot data, and configuration data managed by the Hypervisor no longer retrievable or usable through the interfaces and functions provided by the Hypervisor.

NOTE: This requirement applies to images, snapshot data, and configuration data stored, managed, or exposed as part of the Hypervisor product. It does not extend to copies of such data held exclusively by components outside the declared product boundary or outside the responsibility of the manufacturer, such as external storage systems or backup services not managed by the Hypervisor.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-H-DECOM-001	x	x	x

5.2.2 Management and Orchestration System Requirements

5.2.2.1 General applicability

These requirements apply to the Management and Orchestration (M&O) system regardless of its deployment model. The M&O system may be deployed on dedicated physical hardware, on a dedicated management cluster, or as a virtual machine running on the same virtualised infrastructure that it manages. When deployed as a virtual machine, it is hosted on a management-designated host or cluster segment, and separation from tenant or workload domains is maintained through logical isolation mechanisms such as network segmentation, access control policies, and the CPU, memory, I/O, and integrity isolation mechanisms provided by the hypervisor.

Where the declared product architecture delegates Hypervisor management, administration, lifecycle, configuration, update, logging, or policy enforcement functions to the M&O system, the corresponding Hypervisor requirements may be satisfied collectively by the Hypervisor and the M&O system, provided that the delegated functions, interfaces, security boundaries, and assessment evidence are identified in the product security documentation.

5.2.2.2 Authentication

Classes are cumulative

Basic

REQ-M&O-AUTH-001: The M&O system shall ensure that all administrative interfaces require authentication using unique, non-default credentials.

Elevated

REQ-M&O-AUTH-002: The M&O system shall support multi-factor authentication (MFA) for administrative access and shall enforce the use of revocable authentication mechanisms for administrative and service API access. The M&O system shall also ensure that internal component communication is authenticated.

Advanced

REQ-M&O-AUTH-003: The M&O system shall support integration with external identity and trust services to validate user and service identities and shall support automated credential rotation and revocation where available.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-AUTH-001	x	x	x
REQ-M&O-AUTH-002		x	x
REQ-M&O-AUTH-003			x

5.2.2.3 Authorization

Classes are cumulative

Basic

REQ-M&O-AUTHZ-001: The M&O system shall restrict all administrative actions to authenticated accounts and enforce a default-deny policy for all resource access and control functions.

Elevated

REQ-M&O-AUTHZ-002: The M&O system shall enforce granular authorization for administrative users and service accounts, binding permitted actions to the authenticated identity and enforcing least-privilege permissions by default. This may be implemented using role-based, attribute-based, policy-based, or equivalent access-control mechanisms.

Advanced

REQ-M&O-AUTHZ-003: The M&O system shall enforce fine-grained, policy-driven authorization for administrative users and service accounts, including support for separation of duties and context-aware enforcement where applicable.

The M&O system shall support integration with external identity, access-management, or policy decision systems where such integration is part of the declared product configuration.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-AUTHZ-001	x	x	x
REQ-M&O-AUTHZ-002		x	x
REQ-M&O-AUTHZ-003			x

5.2.2.4 Secure Configuration

Classes are cumulative

Basic

REQ-M&O-CFG-001: The M&O system shall be delivered with secure default settings, including least-privilege access roles, disabled unused services or ports, and enforced secure communication protocols.

Elevated

REQ-M&O-CFG-002: The M&O system shall require explicit authorization before applying any configuration changes to the hypervisor or managed workloads. It shall also require explicit authorization before applying any changes to its own internal configuration.

Advanced

REQ-M&O-CFG-003: The M&O system shall enforce security configuration baselines for all managed hypervisors and its own components and shall automatically detect and alert on any deviation from these baselines.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-CFG-001	x	x	x
REQ-M&O-CFG-002		x	x
REQ-M&O-CFG-003			x

5.2.2.5 Communication Security

Classes are cumulative

Basic

REQ-M&O-COM-001: The M&O system shall protect the confidentiality and integrity of management and orchestration communications exchanged through M&O interfaces and communication paths used to control or manage hypervisors.

Elevated

REQ-M&O-COM-002: The M&O system shall enforce certificate-based mutual authentication for communications used by the M&O system to exchange management or orchestration data with managed hypervisors.

Advanced

REQ-M&O-COM-003: The M&O system shall ensure that control operations are performed only over cryptographically protected and mutually authenticated communication channels and only toward managed hypervisors whose identity has been successfully verified and authorized by the M&O system. The M&O system shall reject control operations when the identity of the addressed managed hypervisor cannot be established or is not authorized.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-COM-001	x	x	x
REQ-M&O-COM-002		x	x
REQ-M&O-COM-003			x

5.2.2.6 Integrity Protection

Classes are cumulative

Basic

REQ-M&O-INT-001: The M&O system shall verify the integrity of orchestration artefacts, including templates, playbooks, declarative GitOps manifests, infrastructure-as-code definitions reconciled by pull-based controllers, and scripts, at import or before execution by validating them against a trusted reference value, such as a cryptographic digest, digital signature, or equivalent integrity reference. For pull-based reconciliation flows, the verification shall be performed at the reconciliation point prior to applying state to managed targets.

Elevated

REQ-M&O-INT-002: The M&O system shall verify the integrity and authenticity of all orchestration artefacts using digital signatures validated against trusted keys, certificates, or equivalent trust anchors.

Advanced

REQ-M&O-INT-003: The M&O system shall validate the integrity and authenticity of all third-party plugins, extensions, or integration modules before activation, and shall support administrative functions to register and revoke trusted signing keys.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-INT-001	x	x	x
REQ-M&O-INT-002		x	x
REQ-M&O-INT-003			x

5.2.2.7 Logging

Classes are cumulative

Basic

REQ-M&O-LOG-001: The M&O system shall log all administrative actions, including user logins, configuration changes, VM lifecycle events, orchestration workflows, and security-related alerts.

Elevated

REQ-M&O-LOG-002: The M&O system shall protect stored logs against tampering and support secure export to external logging or SIEM systems using encrypted, authenticated protocols.

Advanced

REQ-M&O-LOG-003: The M&O system shall support cryptographic signing and time-stamping of audit logs to ensure verification of origin and integrity during forensic analysis.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-LOG-001	x	x	x
REQ-M&O-LOG-002		x	x
REQ-M&O-LOG-003			x

5.2.2.8 Secure Update

Classes are cumulative

Applicability of REQ-M&O-UPD-001 to REQ-M&O-UPD-003:

These requirements apply to updates of the Management and Orchestration (M&O) System, and to update orchestration of managed Hypervisors where that function is part of the declared product or under the responsibility of the manufacturer.

Where update delivery, verification, installation, rollback protection, or update orchestration is performed by a package manager, Host OS update service, platform update service, appliance update mechanism, external lifecycle-management system, or other Operational Environment mechanism outside the declared product or outside the responsibility of the manufacturer, that mechanism shall be treated as an Operational Environment dependency. The product security documentation shall identify the expected update mechanism, the expected security properties, and any assumptions required for secure update.

Basic

REQ-M&O-UPD-001: The M&O system shall be delivered in a form that supports authenticity and integrity verification of M&O updates before installation or activation, using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.

Where the M&O system includes the mechanism that verifies, installs, activates, distributes, or orchestrates updates, this mechanism shall prevent installation, activation, distribution, or orchestration of an update when authenticity or integrity verification fails and shall record a security-relevant event.

If update installation is delegated to an external system that is part of the declared product, the M&O system shall verify and record update provenance and shall record update installation state for updates applied to itself or orchestrated by it for managed hypervisors, based on information exposed by that external system.

Elevated

REQ-M&O-UPD-002: The M&O system shall support rollback protection for M&O updates where rollback protection is provided by the M&O system or by an update mechanism under the responsibility of the manufacturer.

Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed update versions shall be prevented.

Authorized rollback to a previously trusted version shall only be permitted when the rollback image or package is verified for integrity and authenticity, the rollback action is explicitly authorized by an administrator with appropriate privileges or an equivalent authorized control function, and the rollback operation is recorded as a security-relevant event.

Advanced

REQ-M&O-UPD-003: Where the M&O system orchestrates update actions across managed targets as part of the declared product, it shall support secure update orchestration that provides, for each managed target, identification of the update action, verification of update outcome, and detection of failed, partial, or inconsistent rollout states.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-UPD-001	x	x	x
REQ-M&O-UPD-002		x	x
REQ-M&O-UPD-003			x

5.2.2.9 Decommissioning and Secure Transfer

Single Class

Basic

REQ-M&O-DECOM-001: The M&O system shall provide a mechanism to remove orchestration data, management configuration, and associated metadata managed by the M&O system, such that this data is no longer retrievable or usable through the interfaces and functions provided by the M&O system.

REQ-M&O-DECOM-002: Where the M&O system supports transfer of orchestration data, management configuration, or associated metadata to another product or system, the M&O system shall protect the confidentiality and integrity of the transferred data through the interfaces and functions provided by the M&O system.

Requirement applicability by use case and security profile

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-M&O-DECOM-001	x	x	x
REQ-M&O-DECOM-002	x	x	x

5.3 CES Security Requirements

5.3.1 CRS Requirements

5.3.1.1 Isolation

5.3.1.1.1 Container Isolation

Classes are cumulative

Basic

REQ-CRS-CN-ISO-001: The CRS shall enforce isolation between containers (workloads) using operating-system isolation mechanisms under its control, including at least process, filesystem, network, and resource management layers.

The CRS shall support configurations in which one container cannot read, write, or otherwise interfere with another container's processes, filesystem contents, network endpoints, or allocated resources.

Elevated

REQ-CRS-CN-ISO-002: The CRS shall support configuration of strong separation between containers and the host operating system by restricting container access to kernel interfaces, privileged operations, and system resources outside their assigned execution context.

This shall include the capability to:

- limit container privileges through reduction of kernel capabilities and prevention of direct access to host-level device nodes and sensitive filesystems; and
- restrict container access to kernel interfaces and system calls that are not required for the declared workload, using mechanisms, for example, syscall filtering, mandatory access control policies, or equivalent controls.

Advanced

REQ-CRS-CN-ISO-003: The CRS shall support execution of containers within hardened isolation environments that strengthen separation between container workloads and the host kernel, so that compromise of a container in such an environment does not directly provide an attacker with equivalent access to the host kernel or to other containers.

NOTE 1: Hardened isolation environments may be implemented using, for example, micro-VM based container runtimes, user-space kernel sandboxes, or additional Virtualisation or sandboxing layers.

NOTE 2: REQ-CRS-CN-ISO-003 does not require all containers to use hardened isolation by default. It requires that the CRS be capable of providing such isolation for workloads where it is needed.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-CN-ISO-001	x	x	x
REQ-CRS-CN-ISO-002		x	x
REQ-CRS-CN-ISO-003			x

5.3.1.1.2 Control Plane Isolation

Classes are cumulative

Basic

REQ-CRS-CP-ISO-001: The CRS shall support configurations that provide access-control mechanisms for its management and control interfaces. These configurations shall enable containerized workloads to be prevented from directly accessing or invoking CRS management or control functions, and shall ensure that only authenticated and authorized entities can perform control or configuration actions.

CRS management sockets, control APIs, and management daemons shall be deployable in a manner where they are not reachable from within container namespaces.

Elevated

REQ-CRS-CP-ISO-002: The CRS shall support configuration of dedicated logical separation between CRS management and control traffic and container data-plane traffic. The CRS shall support the use of host-only communication endpoints, distinct network namespaces, or dedicated management networks for its management and control interfaces, enabling isolation of CRS control traffic from container workloads.

Advanced

REQ-CRS-CP-ISO-003: The CRS shall support the use of cryptographically protected and mutually authenticated communication channels for all remote or network-exposed CRS management and control interfaces, in order to reduce the risk of unauthorized access, interception, tampering, or lateral movement through the control plane.

Where such channels are used, the CRS shall ensure that management and control traffic is transmitted only over communication paths that provide confidentiality, integrity protection, and mutual authentication between the CRS and authorized management entities.

NOTE: For CRS management interfaces that are restricted to local host-only communication mechanisms with appropriate operating-system access controls, cryptographic protection at the transport layer may not be required, provided that equivalent protection of confidentiality, integrity, and authenticity is achieved through the underlying platform mechanisms.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-CP-ISO-001	x	x	x
REQ-CRS-CP-ISO-002		x	x
REQ-CRS-CP-ISO-003			x

5.3.1.1.3 Network Plane Isolation

Classes are cumulative

Basic

REQ-CRS-NP-ISO-001: The CRS shall maintain logical separation between container network traffic and host or CRS management network traffic, using virtual networking mechanisms under its control.

The CRS shall support configurations in which containers cannot directly access host or CRS management interfaces or services by default, and in which separate network contexts can be defined for management traffic and container workload traffic.

Elevated

REQ-CRS-NP-ISO-002: The CRS shall support configuration of explicit traffic filtering and access-control rules for network flows between containers, between container networks, and between container networks and host or management networks.

These rules shall be enforced at the CRS virtual networking layer or at other enforcement points under the control of the CRS, ensuring that only network flows explicitly permitted by the configured policy are allowed to cross these boundaries.

Advanced

REQ-CRS-NP-ISO-003: The CRS shall support the use of cryptographically protected and mutually authenticated communication channels for management and control traffic that traverses shared or untrusted networks, in order to maintain separation between management traffic and container data-plane traffic.

Where such channels are used, the CRS shall ensure that management and control traffic is transmitted only over communication paths that provide confidentiality, integrity protection, and mutual authentication between the CRS and authorized management entities.

NOTE: For management interfaces that are restricted to local host-only communication mechanisms with appropriate operating-system access controls, cryptographic protection at the transport layer may not be required, provided that equivalent protection of confidentiality, integrity, and authenticity is achieved through the underlying platform mechanisms.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-NP-ISO-001	x	x	x
REQ-CRS-NP-ISO-002		x	x
REQ-CRS-NP-ISO-003			x

5.3.1.2 Integrity Protection

5.3.1.2.1 Boot integrity verification

Single Class

Applicability of REQ-CRS-B-INT-002:

This requirement applies to CRS executable artefacts delivered as part of the declared CRS product.

It does not require the CRS to verify firmware, platform bootloaders, Host OS components, hardware roots of trust, service managers, package managers, or other Operational Environment components, unless those components are part of the declared CRS product.

Where verification is performed by a Host OS, platform-provided boot mechanism, service manager, package manager, appliance mechanism, orchestration mechanism, or other Operational Environment mechanism, that mechanism shall be treated as an Operational Environment dependency. The product security documentation shall identify the expected verification mechanism, the expected security properties, and any assumptions required for the CRS to be securely started or loaded.

Elevated

REQ-CRS-B-INT-002: The CRS shall support deployment with a startup, loading, or installation mechanism that verifies the integrity and authenticity of its executable artefacts before those artefacts are used to enable container scheduling, container lifecycle operations, or container execution interfaces.

The CRS shall be delivered with, or support the use of, verification metadata for those artefacts, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.

Where the CRS includes the mechanism that loads, activates, or starts its executable artefacts, this mechanism shall prevent loading, activation, or startup of those artefacts when verification fails.

Where verification status from a platform-provided startup, loading, or installation mechanism is made available to the CRS, the CRS shall use that status before enabling container scheduling, container lifecycle operations, or container execution interfaces.

If the required verification status cannot be established or is reported as failed, the CRS shall prevent enabling container scheduling, container lifecycle operations, or container execution interfaces where such enforcement is under the control of the CRS.

NOTE 1: CRS executable artefacts may include, depending on the CRS architecture, the container runtime binary, runtime daemon, plugins, or equivalent executable artefacts delivered as part of the CRS.

NOTE 2: Startup, loading, or installation mechanisms relevant to this requirement may include package manager verification, service manager startup integrity checks, appliance startup mechanisms, or equivalent mechanisms by which CRS executable artefacts are loaded or started. Where such mechanisms are provided by the Host OS, platform, or orchestration environment rather than by the CRS itself, they are treated as Operational Environment dependencies. This requirement does not require the CRS to provide those mechanisms unless they form part of the declared CRS product.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-B-INT-002		x	x

5.3.1.2.2 Container image integrity verification

Classes are exclusive

Basic

REQ-CRS-IMG-INT-001: The CRS shall support integrity verification of container images by validating the image manifest and referenced layers against trusted integrity metadata, such as declared cryptographic digests, trusted reference values, or equivalent integrity metadata.

The CRS shall perform, or cause to be performed, this verification at image ingestion, before first use, or before container instantiation according to the configured image trust policy.

If an integrity mismatch, corruption, or missing referenced layer or manifest entry is detected, the CRS shall reject the image, prevent container instantiation from that image where such enforcement is under the control of the CRS, and record a security-relevant event.

NOTE 1: REQ-CRS-IMG-INT-001 focuses on integrity verification of container images using trusted reference values, such as cryptographic digests declared in image manifests or registries.

Elevated

REQ-CRS-IMG-INT-002: The CRS shall implement mechanisms to verify both the integrity and the authenticity of container images before execution. This verification shall establish that each image originates from a trusted publisher or approval authority and has not been altered after it was approved for deployment.

Authenticity verification shall rely on cryptographic mechanisms that provide a verifiable binding between the image and the trusted publisher or approval authority and that prevent undetected modification of the image.

The CRS shall prevent execution or instantiation of any container image whose integrity or authenticity cannot be successfully verified according to the configured trust policy.

NOTE 2: Authenticity verification may be implemented using public-key digital signatures validated via certificate chains rooted in trusted authorities, Sigstore / Cosign-style keyless signing bound to OIDC publisher identities, keyless signing mechanisms bound to identified publishers, or equivalent cryptographic schemes that provide verifiable image provenance and tamper detection.

Advanced

REQ-CRS-IMG-INT-003: The CRS shall perform integrity and authenticity verification of container images using trust anchors stored as protected trust material that is not modifiable during normal operational state.

The protected trust material shall not be modifiable by container workloads, unprivileged users, or routine runtime operations, and shall only be updated or replaced by authenticated and authorized administrators with appropriate privileges and recorded in security-relevant logs sufficient to support security auditing.

If verification of the integrity or authenticity of a container image using this protected trust material fails, or if the verification status of the image cannot be established, the CRS shall prevent execution or instantiation of that container image and shall record a security relevant event.

NOTE 3: Protected trust material refers to cryptographic keys, certificates, or other trust anchors used for container image integrity and authenticity verification, stored and managed so that they cannot be altered through unauthorized administrative or workload facing interfaces. Updates to protected trust material are permitted only via authenticated and authorized administrative actions and are logged for security audit purposes.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-IMG-INT-001	x		
REQ-CRS-IMG-INT-002		x	
REQ-CRS-IMG-INT-003			x

5.3.1.2.3 Runtime integrity protection

Single Class

Applicability of REQ-CRS-RP-INT-002

Runtime integrity protection may rely on mechanisms provided by the CRS, other components of the declared product, or the Operational Environment. Operational Environment dependencies shall be identified in the product security documentation.

Elevated

REQ-CRS-RP-INT-002: The CRS shall support integrity protection for its security-critical runtime code, configuration state, and control logic used to enforce security policies or isolation.

The CRS shall support mechanisms ensuring that security-critical configuration state and control logic can be modified only through authenticated and authorized interfaces or processes.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-RP-INT-002		x	x

5.3.1.2.4 Remote attestation

Single Class

Applicability of REQ-CRS-RA-INT-003 and REQ-CRS-RA-INT-004:

REQ-CRS-RA-INT-003 and REQ-CRS-RA-INT-004 are optional Advanced-class requirements. They apply only when the declared product configuration includes remote attestation as an optional capability for UC-C3 / SP3.

Advanced

REQ-CRS-RA-INT-003: Where remote attestation is included in the declared product configuration, the CRS shall provide generation of attestation evidence describing launch-time measurements of its security-critical executable artefacts.

This attestation evidence shall be cryptographically verifiable by an authorized verifier and bound to the CRS instance for which the measurements were produced.

The CRS shall protect attestation keys and measurement data against unauthorized access, disclosure, or replay.

The attestation evidence shall be limited to the minimum information necessary to verify the integrity of the measured CRS executable artefacts.

Where attestation evidence relies on platform, Host OS, firmware, hardware root of trust, or external attestation services outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms shall be treated as Operational Environment dependencies. The product security documentation shall identify the dependency and the expected security properties.

NOTE 1: This requirement covers the CRS capability to collect, use, bind, or expose launch-time measurements as attestation evidence. The design and operation of local or remote verifiers, attestation protocols, and relying parties is outside the scope of the present document.

NOTE 2: CRS executable artefacts may include, depending on the CRS architecture, the container runtime engine, runtime daemon, runtime shim, execution service, enforcement modules, plugins, helper binaries, or equivalent executable artefacts delivered as part of the CRS.

NOTE 3: This requirement does not require attestation of CRS runtime configuration state, runtime policy state, arbitrary CRS memory, container memory, container workload state, transient execution state, workload-dependent runtime state, or other dynamic runtime data.

REQ-CRS-RA-INT-004: Where attestation evidence produced by the CRS is intended to be used in remote verification scenarios, the CRS shall support configurable privacy-preserving options, as appropriate to the attestation model, such as limiting attested fields to the minimum necessary for integrity verification, using pseudonymous or rotating attestation identifiers, separating platform identity information from integrity measurement information, configuring attestation trigger conditions and frequency, and restricting attestation access to authorized verifiers.

NOTE 4: REQ-CRS-RA-INT-004 does not require specific attestation protocols or identity systems. It requires that, where remote attestation is used, the CRS provides configuration options that allow the operator to balance integrity assurance and privacy considerations.

NOTE 5: Remote attestation is optional in the Advanced class. A product may conform to Advanced without including remote attestation in the declared product configuration. Where remote attestation is not included in the declared product configuration, this limitation is identified in the product security documentation.

Requirement applicability by use case and security profile

NOTE 6: In the following table, "o" indicates an optional capability within UC-C3 / SP3. Absence of support does not preclude Advanced-class conformance.

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-RA-INT-003			o
REQ-CRS-RA-INT-004			o

5.3.1.3 Authentication

5.3.1.3.1 General

This clause distinguishes between authentication of administrative users and authentication of external services that access CRS management functions or data.

5.3.1.3.2 Administrative Authentication

Classes are cumulative

Basic

REQ-CRS-ADMIN-AUTH-001: The CRS shall require authentication for access to all administrative interfaces, including local command-line interfaces, remote application programming interfaces, and administrative dashboards or consoles.

The CRS shall enforce the use of unique, non-default credentials for administrative accounts. Where password-based authentication is supported, the CRS shall enforce password complexity requirements and a mechanism limiting repeated failed authentication attempts, with configurable parameters to mitigate brute-force attempts.

Where password-based authentication is used, the CRS shall store administrative passwords only using non-reversible, salted hashing functions. Default credentials, if present, shall be disabled or shall be changed before administrative access is granted.

Elevated

REQ-CRS-ADMIN-AUTH-002: The CRS shall require authenticated access to administrative interfaces and shall support cryptographic key-based authentication for administrative accounts. The CRS shall allow administrative password-based login to be disabled for administrative interfaces.

- The CRS shall ensure that private keys used for administrative authentication are stored and managed in a manner that prevents unauthorized extraction or reuse by container workloads or other unprivileged components.
- Algorithms and key lengths shall conform to current cryptographic best practices, aligned with Annex K.

Advanced

REQ-CRS-ADMIN-AUTH-003: The CRS shall require strong authentication (as defined in clause 3.1) for all administrative interfaces.

Where certificates are used for administrative authentication, their validity, expiration, and revocation status shall be verified before administrative access is granted.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-ADMIN-AUTH-001	x	x	x
REQ-CRS-ADMIN-AUTH-002		x	x
REQ-CRS-ADMIN-AUTH-003			x

5.3.1.3.3 Service Authentication

Classes are cumulative

Basic

REQ-CRS-SERV-AUTH-001: The CRS shall require authentication for external services accessing CRS application programming interfaces or control interfaces, including orchestration systems, monitoring tools, and backup agents.

The CRS shall enforce the use of distinct credentials for each service account and shall not allow built-in shared or hard-coded service credentials.

The CRS shall provide configurable lockout or rate-limit mechanisms for service authentication attempts and shall store service credentials in a manner that prevents exposure through configuration files, container images, or logs.

Elevated

REQ-CRS-SERV-AUTH-002: The CRS shall support mutual cryptographic authentication for service-to-CRS interactions that access management functions or data, allowing both the CRS and the external service to verify each other's identity before exchanging management or control information.

Cryptographic materials used for service authentication, including keys, certificates, or tokens, shall be protected from unauthorized extraction or reuse by container workloads or other unprivileged components.

Algorithms and key lengths shall conform to current cryptographic best practices, aligned with Annex K.

Advanced

REQ-CRS-SERV-AUTH-003: The CRS shall enforce verifiable binding of service identities to cryptographic credentials for all service interactions that access management functions or data. Service credentials shall be issued, rotated, and revoked through trusted credential or identity management systems, and the CRS shall automatically reject revoked or expired credentials during authentication.

The CRS shall support integration with external identity or credential management systems in order to provide traceability of service identities across deployments and to ensure that only services with valid, managed identities can invoke CRS management or control functions.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-SERV-AUTH-001	x	x	x
REQ-CRS-SERV-AUTH-002		x	x
REQ-CRS-SERV-AUTH-003			x

5.3.1.4 Authorization

5.3.1.4.1 General

This clause distinguishes between authorization of administrative users and authorization of external services that access CRS management functions or data.

NOTE: In this clause, granular authorization means that permitted actions are bound to authenticated identities and least-privilege permissions by default. Fine-grained, policy-driven authorization means that decisions can use more detailed policy rules, separation-of-duties constraints, contextual attributes, or integration with external policy decision systems where supported by the declared product configuration.

5.3.1.4.2 Administrative Authorization

Classes are cumulative

Basic

REQ-CRS-ADMIN-AUTHZ-001: The CRS shall restrict all administrative actions to authenticated accounts and shall enforce a default-deny policy for access to management functions, permitting only those actions that are explicitly authorized.

Elevated

REQ-CRS-ADMIN-AUTHZ-002: The CRS shall enforce granular authorization for administrative accounts, binding permitted actions to the authenticated identity and enforcing least-privilege permissions by default. This may be implemented using role-based, attribute-based, policy-based, or equivalent access-control mechanisms.

Advanced

REQ-CRS-ADMIN-AUTHZ-003: The CRS shall enforce fine-grained, policy-driven authorization for administrative accounts, including support for separation of duties and context-aware enforcement where applicable. The CRS shall support integration with external identity, access-management, or policy decision systems where such integration is part of the declared product configuration.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-ADMIN-AUTHZ-001	x	x	x
REQ-CRS-ADMIN-AUTHZ-002		x	x
REQ-CRS-ADMIN-AUTHZ-003			x

5.3.1.4.3 Service Authorization

Classes are cumulative

Basic

REQ-CRS-SERV-AUTHZ-001: The CRS shall enforce a default-deny policy for all service accounts, ensuring that service accounts are only permitted to perform actions that are explicitly authorized by CRS configuration or policy.

Elevated

REQ-CRS-SERV-AUTHZ-002: The CRS shall enforce granular authorization for service accounts, binding permitted actions to the authenticated service identity and enforcing least-privilege permissions for each service account.

Advanced

REQ-CRS-SERV-AUTHZ-003: The CRS shall enforce fine-grained, policy-driven authorization for service accounts, binding permissions to cryptographic service identities and supporting context-aware enforcement.

The CRS shall support integration with external authorization systems, where such integration is part of the declared product configuration, to enable centralized management and evaluation of authorization policies for service accounts.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-SERV-AUTHZ-001	x	x	x
REQ-CRS-SERV-AUTHZ-002		x	x
REQ-CRS-SERV-AUTHZ-003			x

5.3.1.5 Confidentiality Protection

Classes are cumulative

Basic

REQ-CRS-CONF-001 (Access control for sensitive data): The CRS shall provide access-control mechanisms to restrict access to sensitive data under its control, including container images, configuration data, stored secrets, and security-relevant logs.

Access to such sensitive data shall be permitted only to authenticated and authorized entities through CRS-controlled interfaces and APIs.

NOTE 1: Sensitive data includes, for example, container images stored in registries under CRS control, configuration objects that affect workload behaviour or security, secrets such as credentials and keys managed via the CRS, and logs that may contain security-relevant information.

Elevated

REQ-CRS-CONF-002 (Data at rest): The CRS shall enforce encryption of sensitive data at rest for data that it stores or manages, including container images, configuration data, stored secrets, and security-relevant logs, either through native capabilities or through use of encrypted storage, registries, or secret-management services.

Encryption keys used for data-at-rest protection shall be stored and managed in a manner that prevents access by container workloads and other unprivileged components.

NOTE 2: In multi-tenant deployments, protection of data-at-rest keys may be strengthened through tenant-specific key-management models, such as Bring Your Own Key (BYOK), Hold Your Own Key (HYOK), or equivalent approaches providing comparable tenant separation and key-control assurances.

REQ-CRS-CONF-003 (Data in transit): The CRS shall support the use of encrypted communication for CRS management and control interfaces, for communication with image registries, and for other CRS APIs that handle sensitive data or control operations.

When these communications are configured to use encrypted transport, the CRS shall prevent fallback or downgrade to unencrypted or weakly protected channels.

NOTE 3: Encrypted communication may be implemented using TLS or equivalent cryptographic protocols, including integration with service meshes or platform-provided secure communication mechanisms, provided that confidentiality, integrity, and peer authentication are ensured.

Advanced

REQ-CRS-CONF-004 (Data in use - confidential execution): Where the underlying platform provides mechanisms for isolated or protected execution environments, the CRS shall support provisioning and management of execution environments for containerized workloads in which confidential data associated with those workloads is protected, during execution, from observation, inspection, or access through the CRS, the host operating system, and other workloads.

The CRS shall provide configuration and API interfaces to deploy workloads into such protected execution environments and shall support secure provisioning of secrets and key material to those workloads such that the secrets and key material are not exposed outside the protected provisioning path to the host operating system, other workloads, or CRS components not authorized to handle those materials.

NOTE 4: Platform mechanisms for confidential execution may include hardware-mediated execution enclaves, confidential containers, or other hardware or software mechanisms that provide protected memory and processor-state isolation beyond standard container separation.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-CONF-001	x	x	x
REQ-CRS-CONF-002		x	x
REQ-CRS-CONF-003		x	x
REQ-CRS-CONF-004			x

5.3.1.6 Availability and Resilience

Classes are cumulative

Basic

REQ-CRS-AVAIL-001: The CRS shall maintain resource isolation between container workloads and shall protect CRS control-plane processes from resource exhaustion caused by container workloads. Resource isolation shall cover at least CPU, memory, and storage resources.

The CRS shall provide mechanisms such that failure or abnormal termination of an individual container is contained within its execution context and does not, under normal operating conditions and within configured resource limits, compromise operation of the CRS control plane or cause other containers to terminate unexpectedly.

Elevated

REQ-CRS-AVAIL-002: The CRS shall provide automatic recovery mechanisms such that failed containers are restarted based on defined policies without requiring manual intervention. Where supported by the deployment environment, these mechanisms may be used in conjunction with orchestration or clustering systems to restore workloads on the same or an alternative host.

REQ-CRS-AVAIL-003: The CRS shall implement resource scheduling or reservation mechanisms that support configuration of minimum resource allocations for CRS security and management functions, covering at least CPU, memory, and I/O resources.

These mechanisms shall protect the CRS control plane from resource starvation caused by container workloads or external resource contention, ensuring that security and management functions remain able to operate under load.

Advanced

REQ-CRS-AVAIL-004: Where the deployment environment supports workload relocation or continuity-preserving recovery, the CRS shall support mechanisms to reduce interruption of selected container workloads following control-plane or host-level failures, such as restart from a trusted state, integration with orchestrator recovery, checkpoint and restore where supported, runtime-state preservation, or equivalent continuity mechanisms.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-AVAIL-001	x	x	x
REQ-CRS-AVAIL-002		x	x
REQ-CRS-AVAIL-003		x	x
REQ-CRS-AVAIL-004			x

5.3.1.7 Logging

Classes are cumulative

Basic

REQ-CRS-LOG-001: The CRS shall generate time-stamped audit logs for security-relevant administrative actions and security events, including at least authentication attempts to CRS administrative or control interfaces, configuration changes affecting CRS behaviour or security posture, and container lifecycle operations (creation, start, stop, deletion, and restart) initiated through the CRS.

Each audit record shall include, as applicable to the event type, an event type identifier, the time of the event, the outcome of the event, and the identity of the associated user, service, or process.

The CRS shall ensure that audit logs are not writable or directly modifiable by container workloads or other unprivileged processes running within container namespaces.

NOTE: The set of administrative actions and security events that are considered security-relevant may vary depending on the CRS implementation and its exposed interfaces. The product security documentation should identify which actions/events are treated as security-relevant for audit logging and explain any product-specific additions beyond the minimum set listed above.

Elevated

REQ-CRS-LOG-002: The CRS shall protect audit logs from unauthorized access, modification, or deletion. Access to CRS audit logs shall be restricted to authorized administrative roles or trusted system processes under the control of the CRS or host operating system.

The CRS shall support secure export of audit logs to external log management or security information and event management systems using authenticated and encrypted communication channels.

Where the CRS exposes tenant-visible logs or log-derived information in multi-tenant environments, it shall maintain separation between operator audit logs and tenant-visible logs or telemetry.

Advanced

REQ-CRS-LOG-003: The CRS shall support cryptographic protection, trusted time-stamping, and tamper-evident protection of audit logs to enable verification of the origin and integrity of audit records during forensic analysis. Where the CRS relies on external time or signing services, audit records shall include sufficient metadata to allow verification of the trust source used for time-stamping and integrity protection.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-LOG-001	x	x	x
REQ-CRS-LOG-002		x	x
REQ-CRS-LOG-003			x

5.3.1.8 Secure Update

Classes are cumulative

Applicability of REQ-CRS-UPD-001 to REQ-CRS-UPD-003

These requirements apply to updates of CRS components delivered as part of the declared CRS product.

Where update delivery, verification, installation, rollback protection, or coordinated update is performed by a Host OS package manager, platform update service, appliance update mechanism, orchestration mechanism, lifecycle-management system, or other Operational Environment mechanism outside the declared CRS product or outside the responsibility of the manufacturer, that mechanism shall be treated as an Operational Environment dependency. The product security documentation shall identify the expected update mechanism, the expected security properties, and any assumptions required for secure update of the CRS.

Where update functions are provided collectively by the CRS and other components under the responsibility of the manufacturer, the product security documentation shall identify the responsible components, interfaces, security boundaries, and assessment evidence.

Basic

REQ-CRS-UPD-001: The CRS shall be delivered in a form that supports authenticity and integrity verification of CRS updates before installation or activation, using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.

Where the CRS includes the mechanism that verifies, installs, or activates its updates, this mechanism shall prevent installation or activation of an update when authenticity or integrity verification fails and shall record a security-relevant event.

Elevated

REQ-CRS-UPD-002: The CRS shall support rollback protection for CRS updates where rollback protection is provided by the CRS or by an update mechanism under the responsibility of the manufacturer.

Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed CRS updates shall be prevented.

Authorized rollback to a previously trusted CRS version shall only be permitted when the rollback image or package is verified for integrity and authenticity, the rollback action is explicitly authorized by an administrator with appropriate privileges or an equivalent authorized control function, and the rollback operation is recorded as a security-relevant event.

Advanced

REQ-CRS-UPD-003: Where the CRS supports coordinated update of CRS components across nodes as part of the declared product, the CRS shall provide, for each managed node, identification of the update action, verification of update outcome, and detection of failed or partial rollout states.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-UPD-001	x	x	x
REQ-CRS-UPD-002		x	x
REQ-CRS-UPD-003			x

5.3.1.9 Secure Configuration and Default

Classes are cumulative

Basic

REQ-CRS-CFG-001: The CRS shall, by default, disable interfaces and features that are not required for container execution and CRS administration, including remote debugging endpoints, unauthenticated control sockets, and developer interfaces that expose management or control capabilities.

Only administrators shall be able to explicitly enable such interfaces and features through CRS configuration.

Elevated

REQ-CRS-CFG-002: The CRS shall validate configuration parameters under its control before applying them and shall reject settings that would disable or bypass security mechanisms or that would compromise container-to-container isolation or controlled access to host resources.

When a configuration change is rejected for security reasons, the CRS shall provide clear feedback to the administrator indicating the reason for rejection.

Advanced

REQ-CRS-CFG-003: The CRS shall support the definition and use of security configuration baselines that define secure settings for CRS daemon configuration and default container security profiles, including at least isolation-related options and default access to host resources as implemented within the CRS product.

The CRS shall provide capabilities to automatically validate current CRS configuration and default container profile settings against a selected security baseline and shall generate alerts or audit events when non-compliant settings are detected.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-CFG-001	x	x	x
REQ-CRS-CFG-002		x	x
REQ-CRS-CFG-003			x

5.3.1.10 Data Minimisation

Single Class

Basic

REQ-CRS-DM-001: The CRS shall limit logging, telemetry, and diagnostic data to information necessary for runtime operation, troubleshooting, and security monitoring. Logs, telemetry, and diagnostic outputs produced by the CRS shall not include plaintext credentials, full cryptographic keys, container memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data, or container workload payload data unless such inclusion is explicitly enabled for a specific debugging purpose by an administrator.

Payload content from container workloads shall not be included in CRS logs, telemetry, or diagnostic outputs by default. Where the CRS supports inclusion of payload content for debugging purposes, this behaviour shall only be enabled through explicit administrative configuration and shall be clearly indicated to the administrator.

REQ-CRS-DM-002: The CRS shall provide administrative controls to disable or restrict the collection of specific categories of log, telemetry, and runtime metric data. These controls shall allow administrators to disable or limit the collection of particular log types, telemetry streams, metrics, or debug payload capture, in order to align data collection with operational and regulatory requirements.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-DM-001	x	x	x
REQ-CRS-DM-002	x	x	x

5.3.1.11 Decommissioning

Single Class

Basic

REQ-CRS-DECOM-001: The CRS shall provide a mechanism to decommission a container that renders the associated runtime state and configuration data managed by the CRS no longer retrievable or usable through the interfaces and functions provided by the CRS.

NOTE: This requirement applies to runtime state and configuration data stored, managed, or exposed as part of the CRS product. It does not extend to copies of such data held exclusively by components outside the CRS management boundary, such as external storage systems, registries, backup services, or orchestration components not managed by the CRS.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CRS-DECOM-001	x	x	x

5.3.2 CE Requirements

5.3.2.1 Isolation

Classes are cumulative

Basic

REQ-CE-ISO-001: The CE shall provide a secure interface to configure and invoke CRS operations for the creation, execution, and termination of containers, ensuring that the underlying CRS enforces strong logical separation consistently across all containers. The CE shall prevent container workloads from accessing host administrative functions through interfaces or configuration options exposed by the CE.

Elevated

REQ-CE-ISO-002: The CE shall enforce this by restricting use of CE options and configuration parameters that would weaken container isolation, including privileged execution, host namespace sharing, and unrestricted host mounts, unless explicitly enabled by an authorized administrator.

Advanced

REQ-CE-ISO-003: The CE shall protect its administrative interface against unauthorized access and shall, where such interface is network-accessible, support separation of that interface from container workload traffic and restriction of access to authorized host processes only.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-ISO-001	x	x	x
REQ-CE-ISO-002		x	x
REQ-CE-ISO-003			x

5.3.2.2 Integrity Protection

Classes are exclusive

Basic

REQ-CE-INTEG-001: The CE shall verify the integrity of container images and associated metadata, including manifests and referenced layers, against trusted integrity metadata before making those images available to the CRS. Images that fail verification shall not be made available to the CRS.

Elevated

REQ-CE-INTEG-002: The CE shall verify the authenticity and integrity of container images using digital signatures validated against trusted keys, certificates, or equivalent trust anchors.

Advanced

REQ-CE-INTEG-003: The CE shall verify the authenticity and integrity of container images using trust anchors stored as protected trust material that is not modifiable during normal operational state. If verification of the integrity or authenticity of a container image using this protected trust material fails, or if the verification status of the image cannot be established, the CE shall prevent that image from being made available to the CRS.

NOTE: Protected trust material refers to keys, certificates, or other trust anchors used for container image integrity and authenticity verification, stored and managed so that they cannot be altered through unauthorized administrative or workload-facing interfaces.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-INTEG-001	x		
REQ-CE-INTEG-002		x	
REQ-CE-INTEG-003			x

5.3.2.3 Authentication

Classes are cumulative

Basic

REQ-CE-AUTH-001: The CE shall require authentication on all interfaces used by orchestration components or external agents to manage containers, ensuring that only authenticated actors can execute workload lifecycle or configuration operations.

Elevated

REQ-CE-AUTH-002: The CE shall enforce cryptographic key-based mutual authentication for communication with the orchestrator or authorized agents. Private keys shall be protected from unauthorized extraction or reuse.

Advanced

REQ-CE-AUTH-003: The CE shall enforce certificate-based mutual authentication for all management interfaces. Certificates used for authentication shall be validated against a trusted public key infrastructure, including verification of certificate validity, expiration, and revocation status, before access is granted.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-AUTH-001	x	x	x
REQ-CE-AUTH-002		x	x
REQ-CE-AUTH-003			x

5.3.2.4 Authorization

Classes are cumulative

Basic

REQ-CE-AUTHZ-001: The CE shall enforce access control on all operations that affect container lifecycle or resource usage, using a default-deny policy and evaluating the authenticated identity of the caller.

Elevated

REQ-CE-AUTHZ-002: The CE shall enforce granular authorization for control operations, binding permitted actions to the authenticated identity and enforcing least-privilege permissions by default. This may be implemented using role-based, attribute-based, policy-based, or equivalent authorization mechanisms.

Advanced

REQ-CE-AUTHZ-003: The CE shall enforce fine-grained, policy-driven or context-aware authorization for control operations, supporting policy decision points and integration with external authorization systems where supported by the declared product configuration.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-AUTHZ-001	x	x	x
REQ-CE-AUTHZ-002		x	x
REQ-CE-AUTHZ-003			x

5.3.2.5 Confidentiality

Classes are cumulative

Basic

REQ-CE-CONF-001 (Protection of stored data): The CE shall prevent unauthorized access to container images, configuration data, and secrets stored or managed by the CE by enforcing CE-controlled access boundaries and ensuring that such data is not exposed to workloads or users unless explicitly authorized through CE configuration or APIs.

Elevated

REQ-CE-CONF-002 (Data at rest and in transit): The CE shall support encryption at rest for container images, configuration data, and secrets under its control, and shall ensure that CE administrative interfaces and communications with external services are protected using encrypted channels.

Advanced

REQ-CE-CONF-003 (Data in use): Where the underlying platform exposes protected execution contexts that provide confidentiality of workload data in use against privileged software (confidential execution), the CE shall provide configuration and API interfaces to enable workload execution in such protected contexts and shall, where the user opts to use these capabilities, enforce placement and associated security policies such that declared confidential workloads run only in such protected contexts. The CE shall support secure provisioning of secrets and key material into the protected execution context such that they are not exposed outside the protected provisioning path to the host operating system, other workloads, or CE components not authorized to handle those materials.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-CONF-001	x	x	x
REQ-CE-CONF-002		x	x
REQ-CE-CONF-003			x

5.3.2.6 Secure Update

Classes are cumulative

Applicability of REQ-CE-UPD-001 and REQ-CE-UPD-002

These requirements apply to updates of CE components delivered as part of the declared CES product

Where update delivery, verification, installation, or rollback protection is performed by a Host OS package manager, platform update service, appliance update mechanism, orchestration mechanism, lifecycle-management system, or other

Operational Environment mechanism outside the declared CE product or outside the responsibility of the manufacturer, that mechanism shall be treated as an Operational Environment dependency. The product security documentation shall identify the expected update mechanism, the expected security properties, and any assumptions required for secure update of the CE.

Basic

REQ-CE-UPD-001: The CE shall be delivered in a form that supports authenticity and integrity verification of CE updates before installation or activation, using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.

Where the CE includes the mechanism that verifies, installs, or activates its updates, this mechanism shall prevent installation or activation of an update when authenticity or integrity verification fails and shall record a security-relevant event.

Elevated

REQ-CE-UPD-002: The CE shall support rollback protection for CE updates where rollback protection is provided by the CE or by an update mechanism under the responsibility of the manufacturer.

Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed CE updates shall be prevented.

Authorized rollback to a previously trusted CE version shall only be permitted when the rollback image or package is verified for integrity and authenticity, the rollback action is explicitly authorized by an administrator with appropriate privileges or an equivalent authorized control function, and the rollback operation is recorded as a security-relevant event.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-UPD-001	x	x	x
REQ-CE-UPD-002		x	x

5.3.2.7 Decommissioning and secure data deletion

Single Class

Basic

REQ-CE-DECOM-001: The CE shall provide a mechanism to remove container images, cached image layers, and associated configuration data managed by the CE that renders such data no longer retrievable or usable through the interfaces and functions provided by the CE.

NOTE: This requirement applies to container images, cached image layers, and configuration data stored, managed, or exposed as part of the CE product. It does not extend to copies of such data held exclusively by components outside the CE management boundary, such as external registries, storage systems, backup services, or orchestration components not managed by the CE.

REQ-CE-DECOM-002: Where the CE supports transfer of container image metadata, configuration, or associated management data to another product or system, the CE shall protect the integrity of the transferred data and prevent unauthorized disclosure or modification during the transfer.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CE-DECOM-001	x	x	x
REQ-CE-DECOM-002	x	x	x

5.3.3 CO Requirements

5.3.3.1 Isolation

Classes are cumulative

Basic

REQ-CO-ISO-001: The CO shall enforce strong logical network isolation between workloads scheduled in different namespaces or tenancy domains, using network isolation policies or equivalent orchestration mechanisms.

Elevated

REQ-CO-ISO-002: The CO shall ensure its management and orchestration interfaces (API server) are logically isolated from the container data plane to prevent unauthorized pivot attempts from compromised workloads.

Advanced

REQ-CO-ISO-003: The CO shall enforce cryptographic protection on control plane communications and ensure that control plane interfaces cannot be accessed from workload or data plane networks.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-ISO-001	x	x	x
REQ-CO-ISO-002		x	x
REQ-CO-ISO-003			x

5.3.3.2 Integrity Protection

Classes are cumulative

Basic

REQ-CO-INTEG-001: The CO shall protect the integrity of its configuration and policy state, ensuring that only authorized components and roles can modify control plane configuration, workload specifications, or cluster policies.

Elevated

REQ-CO-INTEG-002: The CO shall verify the integrity of its control plane component binaries (including API server and scheduler) before operation, using cryptographic digests, digital signatures, or equivalent trusted reference values.

Advanced

REQ-CO-INTEG-003: The CO shall enforce integrity of workload deployments at admission time by applying admission control policies and preventing deployment of workloads that violate integrity, authorization, or configuration constraints.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-INTEG-001	x	x	x
REQ-CO-INTEG-002		x	x
REQ-CO-INTEG-003			x

5.3.3.3 Authentication

Classes are cumulative

Basic

REQ-CO-AUTH-001: The CO shall enforce authenticated access for all administrative and user interactions with the control plane APIs and management interfaces. Where password-based authentication is used, password strength and lockout policies shall be enforced to mitigate brute-force attempts.

Elevated

REQ-CO-AUTH-002: The CO shall enforce cryptographically verifiable authentication mechanisms for service accounts and administrative identities and shall require mutual authentication for internal control plane communication.

NOTE: This may be achieved through mechanisms such as mutual TLS (mTLS), token-based identity, SPIFFE/SPIRE identities, or equivalent cryptographically authenticated identity systems.

Advanced

REQ-CO-AUTH-003: The CO shall enforce certificate-based authentication across all control plane and administrative interfaces and shall support integration with a trusted Public Key Infrastructure (PKI) for issuing and managing these certificates.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-AUTH-001	x	x	x
REQ-CO-AUTH-002		x	x
REQ-CO-AUTH-003			x

5.3.3.4 Authorization

Classes are cumulative

Basic

REQ-CO-AUTHZ-001: The CO shall enforce a default-deny authorization policy for all control plane actions and resource access, restricting actions to authenticated entities.

Elevated

REQ-CO-AUTHZ-002: The CO shall enforce granular authorization for workload and resource control operations, binding permitted actions to authenticated users or service accounts and enforcing least-privilege permissions by default. This may be implemented using role-based, attribute-based, policy-based, or equivalent authorization mechanisms.

Advanced

REQ-CO-AUTHZ-003: The CO shall enforce fine-grained, policy-driven or context-aware authorization for control plane actions, enabling policy enforcement based on runtime context where applicable and integration with external authorization systems where supported by the declared product configuration.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-AUTHZ-001	x	x	x
REQ-CO-AUTHZ-002		x	x
REQ-CO-AUTHZ-003			x

5.3.3.5 Confidentiality

Classes are cumulative

Basic

REQ-CO-CONF-001 (Secrets Protection): The CO shall ensure that control plane secrets, including authentication tokens, private keys, and service credentials, are not stored in plaintext and are accessible only through a controlled interface. The CO shall support either internal secrets store or integration with a Key Management System (KMS).

Elevated

REQ-CO-CONF-002 (Data in transit): The CO shall ensure that control-plane traffic and orchestrator-managed communication channels that it configures or enforces are encrypted. The CO shall enforce configuration policies that prevent the use of unencrypted communication paths for control-plane traffic and for orchestrator-managed communication channels under its control.

Advanced

REQ-CO-CONF-003 (Data in use): The CO shall provide mechanisms to place declared confidential workloads only on nodes that expose the required confidential-execution capabilities and shall enforce the associated placement, isolation, and policy constraints.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-CONF-001	x	x	x
REQ-CO-CONF-002		x	x
REQ-CO-CONF-003			x

5.3.3.6 Secure Update

Classes are cumulative

Applicability of REQ-CO-UPD-001 to REQ-CO-UPD-003

These requirements apply to updates of CO components delivered as part of the declared CES product.

Where update delivery, verification, installation, rollback protection, or cluster update orchestration is performed by a Host OS package manager, platform update service, appliance update mechanism, lifecycle-management system, or other Operational Environment mechanism outside the declared CO product or outside the responsibility of the manufacturer, that mechanism shall be treated as an Operational Environment dependency. The product security documentation shall identify the expected update mechanism, the expected security properties, and any assumptions required for secure update of the CO.

Basic

REQ-CO-UPD-001: The CO shall be delivered in a form that supports authenticity and integrity verification of CO updates before installation or activation, using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
Where the CO includes the mechanism that verifies, installs, or activates its updates, this mechanism shall prevent installation or activation of an update when authenticity or integrity verification fails and shall record a security-relevant event.

Elevated

REQ-CO-UPD-002: The CO shall support rollback protection for CO updates where rollback protection is provided by the CO or by an update mechanism under the responsibility of the manufacturer.

Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed CO updates shall be prevented.

Authorized rollback to a previously trusted CO version shall only be permitted when the rollback image or package is verified for integrity and authenticity, the rollback action is explicitly authorized by an administrator with appropriate privileges or an equivalent authorized control function, and the rollback operation is recorded as a security-relevant event.

Advanced

REQ-CO-UPD-003: Where the CO supports execution of update actions across managed cluster components as part of the declared product, the CO shall provide, for each managed component, identification of the update action, verification of update outcome, and detection of failed or partial rollout states.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-UPD-001	x	x	x
REQ-CO-UPD-002		x	x
REQ-CO-UPD-003			x

5.3.3.7 Decommissioning and secure data deletion

Single Class

Basic

REQ-CO-DECOM-001: The CO shall provide a mechanism to remove workload specifications, orchestration configuration, policy objects, and associated metadata managed by the CO, such that this data is no longer retrievable or usable through the interfaces and functions provided by the CO.

REQ-CO-DECOM-002: Where the CO supports transfer of workload specifications, orchestration configuration, policy objects, or associated metadata to another product or system, the CO shall protect the confidentiality and integrity of the transferred data through the interfaces and functions provided by the CO.

Requirement applicability by use case and security profile

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-CO-DECOM-001	x	x	x
REQ-CO-DECOM-002	x	x	x

5.4 Assurance Security Requirements

5.4.1 Vulnerability Management

REQ-ASSUR-VULM-001: The product shall not contain unmitigated known exploitable vulnerabilities.

NOTE : For the purposes of the present requirement, a known exploitable vulnerability is considered mitigated where appropriate technical or operational measures, including measures described in the product guidance where relevant, sufficiently prevent or reduce the feasibility of exploitation in the product context, taking into account the intended purpose and reasonably foreseeable use of the product.

Requirement applicability by use case and security profile

For VES

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-ASSUR-VULM-001	x	x	x

For CES

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-ASSUR-VULM-001	x	x	x

5.4.2 Software Bill of Materials (SBOM)

REQ-ASSUR-SBOM-001: A Software Bill of Materials (SBOM) shall be provided for VES and CES products, covering the in-scope components of the VES and CES as described in clauses 4.2.2.2 and 4.2.3.1. The SBOM shall identify the software components and dependencies included in the product, including their names, versions, and supplier or origin information where available; for open-source components, Software Heritage persistent Identifiers (SWHIDs) shall be included where available as a means of satisfying the origin information requirement of this clause. The SBOM shall be provided in a machine-readable format.

REQ-ASSUR-SBOM-002: For VES, the SBOM shall include, at minimum:

- Hypervisor executable artefacts, such as the Hypervisor image, kernel, microkernel, virtual machine monitor, or equivalent executable components.
- Hypervisor management interfaces, such as CLI or API components.
- Virtual Machine Monitor (VMM) and lifecycle controllers, where applicable.
- Resource scheduling components for CPU, memory, or I/O, where included in the product.
- Device virtualisation modules.
- Hardware-related dependencies bundled with the VES product.

REQ-ASSUR-SBOM-003: For CES, the SBOM shall include, at minimum:

- CRS executable artefacts, such as runtime binaries, runtime daemons, runtime shims, enforcement modules, APIs, or services.
- Container Engine components, such as engine binaries, image management tools, container networking modules, storage or volume plugins, and supported extensions.
- Container Orchestrator components, such as orchestrator binaries, admission controllers, scheduling modules, secrets management, network policy enforcement, and cluster lifecycle management components.
- Hardware-related dependencies bundled with the CES product.

NOTE: In this clause, examples are provided to illustrate typical components and formats that may fall under the SBOM requirements. These examples are non-exhaustive and non-normative. For instance:

- Device virtualisation modules may include SR-IOV drivers, paravirtualised device drivers, or host-guest interface modules.
- CRS runtime binaries may include runc or crun.
- CRS enforcement modules may include namespaces, seccomp, cgroups, or microVMs.
- CRS APIs/services may include containerd, CRI-O.
- Container engine binaries may include Docker or Podman.
- Container orchestrator binaries may include Kubernetes®.
- Machine-readable SBOM formats may include SPDX or CycloneDX.
- Hardware-related dependencies may include vTPM firmware, drivers for SR-IOV NICs, storage controllers, or other hardware-specific modules.

Requirement applicability by use case and security profile

For VES

Requirements	UC-V1 / SP1	UC-V2 / SP2	UC-V3 / SP3
REQ-ASSUR-SBOM-001	x	x	x
REQ-ASSUR-SBOM-002	x	x	x
REQ-ASSUR-SBOM-003			

For CES

Requirements	UC-C1 / SP1	UC-C2 / SP2	UC-C3 / SP3
REQ-ASSUR-SBOM-001	x	x	x
REQ-ASSUR-SBOM-002			
REQ-ASSUR-SBOM-003	x	x	x

6 Assessment criteria for compliance with technical requirements

6.1 Assessment Methodology

6.1.1 General Assessment Procedure

- This clause defines the methodology for assessing conformity with the VES and CES security requirements specified in the present document.
- The assessment shall be performed according to:
  1) the product's declared configuration and included components as defined in clauses 4.2 and 4.2.5;
  2) the representative use case applicable to the product context;
  3) the threats applicable to the declared deployment and use case; and
  4) the applicable SP1/SP2/SP3 corresponding to the assessed risk level (Low/Medium/High).
- Assessment activities shall verify that the implemented requirements mitigate the threats associated with the declared deployment and use case.
- Each requirement is associated with an assessment description that follows a structured format:
  1) **Assessment reference:** Identifies the link to the exact requirement ID(s).
  2) **Assessment objective:** Defines the security property or capability that shall be verified, ensuring that the assessment remains focused on the intent of the requirement.
  3) **Assessment preparation:** Describes the environment, setup, and preconditions required before executing the test. It includes the following elements as applicable:
     - Test environment: Describe the hardware, software, and network setup used for the assessment, including versions, topology, and any relevant dependencies.
     - Preconditions: Specify any configurations, credentials, or operational states that should be established before the test (e.g. product initialized, certificates loaded, user roles created).
     - Required tools: Identify the tools or software necessary to perform the assessment (e.g. vulnerability scanners, protocol fuzzers, traffic analysers, static code analysers, cryptographic test suites).
     - Reference any vendor-provided setup guides, configuration instructions, or operational manuals, as well as any relevant standards or technical notes, that define how the product shall be configured or operated for the assessment.
     - Product configuration documentation: Confirm that the assessment environment reflects the product composition declared by the manufacturer as defined in clauses 4.2 and 4.2.5, including which components are included or excluded. For example:

- If the product includes only the Hypervisor, M&O components are not assessed.
- If the product includes both the Hypervisor and the M&O system, each component is assessed against its respective requirements, and the assessment shall additionally verify the security-relevant interactions between them.
- If optional components are excluded, the assessment shall verify that their absence does not affect requirement fulfilment.

4) **Assessment activities**: Defines the assessment steps to be performed for the requirement. The specific assessment activities shall be defined in the corresponding requirement-specific assessment case and may include documentation review, configuration inspection, security testing, code or binary analysis, or observation of runtime behaviour, as relevant to the assessed requirement.

5) **Assessment verdict**: Defines the pass/fail criteria.

- Pass:
The assessment is considered passed if the product demonstrably fulfils the requirement and meets any requirement-specific acceptance criteria defined in the corresponding assessment case.

- Fail**:**
The assessment is considered failed if the requirement is not fulfilled, or if the requirement-specific acceptance criteria defined in the corresponding assessment case are not achieved.

6) **Assessment evidence**: Defines the artefacts and documentation that shall be collected to demonstrate that the requirement has been assessed and fulfilled. The evidence shall be sufficient to enable independent verification of the assessment results and to demonstrate compliance with the relevant CRA essential requirements. The assessment evidence includes, where applicable:

- Test or assessment reports showing the steps performed and results obtained;
- Logs, configuration files, or audit traces demonstrating the implementation of the requirement;
- Screenshots, captures, or console outputs confirming the correct execution or protection behaviour;
- Relevant vendor or design documentation describing the applied security measures;
- Evidence of the declared product composition used to scope the assessment (as per clauses 4.2 and 4.2.5);
- Use-case description and risk level justification applied to determine the applicable SP;
- Threat applicability analysis demonstrating which threats from clause B.2 were considered for the assessed configuration;
- Configuration-specific test results where feature or deployment variations affect implementation of the requirement;
- Evidence of security-relevant interactions between in-scope components when multiple components (e.g. Hypervisor and M&O system) are included in the product.

The assessment is considered incomplete if the assessment evidence is missing, inconsistent, or insufficient to verify fulfilment of the requirement.

The conformity assessment defined in the present document is evidence-based. Assessment activities are performed in order to obtain, examine, or confirm the evidence necessary to demonstrate that the product, in its declared configuration, fulfils the applicable requirements.

6.1.2 Technology-neutral Assessment

6.1.2.1 General

This clause complements the general assessment procedure defined in clause 6.1.1. It provides additional guidance for assessing requirements, in particular where such requirements refer to platform-anchored, hardware-assisted, or externally protected trust mechanisms.

6.1.2.2 Equivalence of Hardware-based and Software-based Approaches

For requirements that refer to platform-anchored, hardware-assisted, or externally protected trust mechanisms, products may fulfil the requirement by implementing:

- Hardware-based trust services; or
- Software-based or hybrid mechanisms that provide demonstrably equivalent security outcomes and verifiable assurance.

Where relevant to the requirement being assessed, equivalence shall be demonstrated by showing that the implementation achieves the corresponding security properties listed below:

1) **Cryptographic chain of trust**
   A verifiable cryptographic chain of trust is established from boot through runtime, where each stage validates the integrity and authenticity of the next stage using digital signatures or equivalent integrity verification of authenticated components.

2) **Protected key storage**
   Cryptographic keys and credentials used to enforce trust are managed and stored under strict access-control policies that prevent unauthorized reading, modification, or substitution, including by privileged software components.

3) **Runtime integrity verification**
   Continuous or periodic validation of security-critical components, such as configuration data, binaries, and privileged memory regions, is implemented, with automated detection and documented response to integrity violations.

4) **Attestation capability**
   The product can generate cryptographically verifiable evidence of its configuration and integrity state that can be validated by an authorized verifier.

5) **Transparency of trust anchors and residual assumptions**
   Where the implementation anchors trust in hardware roots of trust, hardware-mediated execution enclaves, hardware security modules, or in protected software-based trust material, the product documentation shall describe the trust anchors and protected trust material on which the implementation depends for boot-chain integrity, image verification, runtime integrity, attestation, or data-in-use protection. Examples of such hardware components may include CPU security features, firmware trust stores, trusted platform modules, trusted execution environments, hardware security modules, secure elements, or equivalent platform trust anchors. The documentation shall identify at least:

   - The hardware and software components that act as trust anchors or provide protected trust material;
   - The security properties that are assumed for these components; and
   - Residual risks if those assumptions do not hold, including the limits of what software-based mechanisms can detect or prevent and potential impacts of hardware-level or supply-chain compromise consistent with the threat scenarios defined in clause B.2 (such as T-ALL-HW-ACCESS).

6) Where key storage or key protection is relevant to the assessed requirement, the expected level of key management protection shall be proportionate to the applicable security profile class:

   - SP1: software-based key management may be used, provided that access controls prevent unauthorized access, modification, or substitution of the relevant keys or credentials.
   - SP2 and SP3: key management shall rely on dedicated key storage mechanisms, which may be hardware-backed or implemented through software-based protected storage, provided that equivalent protection against unauthorized access, modification, or substitution can be demonstrated.

Where a software-based implementation is used to demonstrate comparable protection under this clause, the compensating controls and assessment principles in clause 6.1.2.3 shall be applied in evaluating the sufficiency of that demonstration.

Technology-neutral equivalence shall not be presumed from the presence of compensating controls alone. The assessment shall verify that the alternative implementation achieves the specific security property required by the applicable requirement, considering the relevant threat scenario, attack surface, assumptions, and residual risks.

The assessment shall weigh the totality of evidence against the protection objective of the applicable threat(s).

The assessment shall verify achievement of the intended security outcomes of the relevant requirements, rather than requiring the use of specific hardware technologies. Hardware-based and software-based implementations shall be considered equivalent where they demonstrably provide comparable protection against the applicable threats and offer verifiable assurance of their security properties, taking into account the documented trust assumptions and residual risks. The absence of hardware-based trust mechanisms shall not, by itself, be treated as a deficiency provided that the assessment evidence demonstrates that the implemented software-based or hybrid mechanisms achieve the required security properties and that any residual risks are identified and justified. Software-based implementations without hardware dependencies may be used to meet Advanced-class requirements where they demonstrably achieve comparable protection against the mapped threats.

6.1.2.3 Technology-neutral Assessment Principles

The general assessment methodology defined in clause 6.1.1 applies to all assessments under the present document. The principles in this clause provide additional guidance for assessment of requirements, where technology-neutral equivalence between hardware-based, software-based, or hybrid approaches is relevant.

When applying the methodology in clause 6.1.1 to requirements of any class, the assessment shall:

- Evaluate the implementation against the applicable threats identified in clause B.2 and the declared use case and risk level, rather than against a preferred reference architecture;
- Focus on security outcomes and effectiveness of threat mitigation, not on the presence of particular hardware features or vendor technologies;
- Not classify an implementation as deficient solely because it does not use specific hardware mechanisms, such as hardware roots of trust, hardware-mediated execution enclaves, hardware security modules, IOMMU, or equivalent mechanisms, if the product demonstrates equivalent security properties through alternative architectural means;
- Consider compensating controls such as transparency, auditability, rapid update capability, formal verification or extensive testing results when hardware immutability or hardware trust anchors are not used; and
- Verify that documentation of trust anchors, trust assumptions, and residual risks is available and consistent with the implemented architecture and the applicable requirements.

These principles apply in particular to requirements that reference platform-anchored trust mechanisms for boot integrity, image verification, runtime integrity, remote attestation, or data-in-use protection.

Evidence that a software-based implementation achieves comparable protection may include, without limitation, and consistently with the compensating controls identified in this clause:

a) documented access control architecture preventing unauthorized access to software-managed key material, including privilege separation, process isolation, and audit logging;

b) results of security testing or penetration testing of the key management implementation;

c) formal or semi-formal analysis of the access control properties of the key store;

d) publicly auditable source code permitting independent verification of the key management implementation and its access controls; and

e) demonstrated patch response history for key management vulnerabilities, evidencing rapid update capability.

These compensating controls are relevant in particular to the equivalence determination under clause 6.1.2.2 and to the SP2 and SP3 key management demonstration required under clause 6.1.2.2.

6.1.3 Assessment of RDPS requirements

Where Annex R applies in accordance with clause 5.1, the corresponding assessment provisions specified in Annex R shall be used in addition to the assessment criteria specified in clause 6 for the applicable core requirements of clause 5.

The assessment of Annex R shall identify:

a) the RDPS-dependent product function;

b) the product-facing RDPS interface;

c) the relevant RDPS interaction content;

d) the applicable Annex R requirement family or families;

e) the applicable Annex R requirement level, according to the Security Profile applicable to the RDPS-dependent product function; and

f) any Operational Environment dependency or third-party assurance evidence relied upon for the RDPS boundary.

Where the RDPS side is part of the declared product, provided by the manufacturer, or under the responsibility of the manufacturer for the RDPS-dependent product function, the applicable RDPS-side assessment cases specified in clause R.5.2 shall be assessed using evidence for that RDPS side.

Where the RDPS side is outside the declared product boundary and outside the responsibility of the manufacturer, the assessment shall identify the corresponding Operational Environment dependency, the expected security properties, and available third-party assurance evidence, where relevant.

The assessment of Annex R does not replace the assessment of the applicable core requirements of clause 5. Where an M&O System, CE, or CO is part of the declared VES or CES product, the corresponding assessment cases in clause 6 remain applicable to those components, regardless of whether their implementation is local, distributed, or remotely deployed.

6.2 Assessment Report Requirements

6.2.1 Mandatory Report Contents

The assessment report shall provide sufficient information to demonstrate that the assessment was performed in accordance with the present document, that the scope of the assessment corresponds to the declared product configuration, and that the assessed requirements mitigate the applicable threats.

The report shall include the following clauses:

a) Product Identification and Configuration

- Product name, version, manufacturer, and build or release identifier.
- Declared product composition as defined in clauses 4.2 and 4.2.5, clearly identifying:
  - Components included in the assessed product.
  - Components excluded from the product and therefore not assessed.
- Architectural classification (e.g. Hypervisor Type 1 / Type 2 / Hybrid, Virtual Execution Stack configuration).
- Any declared platform or hardware dependencies relevant to security enforcement.

b) Use Case, Risk, and Requirement Application

- Declared use case(s) and deployment context.
- Risk level (Low / Medium / High) determined in accordance with Annex B, and corresponding Security Profile (SP1/SP2/SP3).

- Applicable requirement set derived from the representative use case applicable to the product context.

c) Threat Applicability

- List of applicable threats from clause B.2 relevant to the declared configuration and use case.

d) Assessment activities and Results

For each assessed requirement, the report shall provide:

- Requirement ID and full requirement text.
- Requirement class (Basic / Elevated / Advanced).
- Linked threat(s) from clause B.2 and rationale for applicability.
- Assessment activities performed, including any class-specific or configuration-specific procedures.
- Test environment and configuration details (e.g. hardware features, firmware level, cluster topology, VM/container layout).
- Observations and results.
- Verdict (Pass / Fail) with justification.
- Assessment evidence Reference (using categories defined in clause 6.2.3).

Where alternative assessment paths are permitted (e.g. due to unavailable platform features), the report shall document:

- The unavailable capability and reason
- The alternative procedure used
- Justification for equivalence of assurance

e) Component-Level and System-Level Findings

- Results for each individual component in the product composition.
- System-level interaction findings, including isolation, trust boundaries, and interface mediation, where multiple components are present.
- Identification of any constraints, operational assumptions, or observed limitations.

f) Overall Conformity Statement

- Statement of conformity or non-conformity with the present document.
- Assigned SP (SP1/SP2/SP3) and the representative use case applicable to the product context.
- Residual risks, limitations, known deviations, or conditional security dependencies.

g) Operational Environment Dependencies for Advanced Class Assessments

When an assessment at the Advanced class relies on Operational Environment (OE) hardware components to achieve the required assurance level, such as hardware roots of trust, hardware-mediated execution enclaves, hardware security modules, or other platform trust anchors, the assessment report shall explicitly document:

- Where applicable, the specific hardware security components including their version and their security properties required for conformity (e.g. secure key storage, measured boot, attestation capability).
- Where applicable, the specific software security components including their version and their security properties required for conformity (e.g. secure key storage, measured boot, attestation capability).
- The manner in which these components contribute to the enforcement of the assessed requirements.
- Any platform prerequisites or firmware dependencies necessary for correct operation.

The report shall reference corresponding evidence under category [PLT-CAP] (Platform Capability Evidence).

6.2.2 Traceability Requirements

The assessment report shall provide complete traceability demonstrating:

Threat → Risk Level → Requirement → Assessment Activity → Evidence → Verdict

This traceability shall be sufficient to enable an independent reviewer to:

- Confirm that all applicable threats were addressed.
- Verify that the rigor of assessment is consistent with the declared SP.
- Reconstruct assessment decisions without direct access to the product.

6.2.3 Evidence Package

The assessment shall be supported by an evidence package referenced by the report. Evidence should be organized using the evidence categories listed in table 6.2.3-1.

Table 6.2.3-1: Evidence categories

Tag	Evidence Category	Purpose	Examples
[LOG]	Execution Logs & Test Outputs	Shows what was done and what happened during testing	Console logs, CLI output, packet captures, fuzz results
[CONFIG]	System & Configuration State	Captures how the system was configured at test time	Config files, ACLs, hypervisor settings, routing tables
[SCOPE]	Product Composition & Scope Evidence	Shows which components are included/excluded per clauses 4.2 and 4.2.5	Product composition statement, SBOM, component list
[RA]	Risk & Threat Applicability Evidence	Shows link to Annex B scoring & threat selection	Risk scoring tables, threat mapping justification
[DESIGN]	Architecture & Implementation Documentation	Explains how the security mechanism works	Security architecture documentation, mitigation mechanism descriptions, trust-model descriptions, component-boundary descriptions, formal verification evidence, security analysis, or equivalent architectural evidence for diverse implementation approaches including, for split-domain or paravirtualised Type I Hypervisor implementations, documentation of the interdomain communication mechanism, including grant tables, event channels, or equivalent channel specifications and associated access controls; documentation of the management domain privilege model and the boundary controls preventing management domain components from directly accessing guest memory or CPU state outside defined interfaces; and documentation of driver domain isolation and the mechanisms by which I/O paths are mediated without direct hardware access from guest domains
[OBSERVATION]	Captured Traces & Measurement Evidence	Instrumentation proving enforcement or behaviour	Screenshots, Wireshark traces, leakage graphs, audit views including, for split-domain or paravirtualised Type I Hypervisor implementations, audit or logging records for interdomain communication events relevant to the assessed security property
[PLT-CAP]	Platform Capability Evidence	Documents feature availability or dependency	CPU/IOMMU features, trust anchor presence, firmware versions, platform-exposed integrity status interfaces, or other platform capability or dependency evidence relevant to the assessed requirement

6.3 VES

6.3.1 Hypervisor

6.3.1.1 Assessment for VM Isolation

6.3.1.1.1 Assessment Case AC-H-VM-ISO-001

1. Assessment reference

Requirement: REQ-H-VM-ISO-001 (Basic)

2. Assessment objective

Verify that the Hypervisor enforces isolation between guest virtual machines at CPU, memory, I/O, and interrupt-handling levels, ensuring that one guest virtual machine cannot read or modify resources of another guest virtual machine, and cannot degrade the availability of another guest virtual machine beyond what is permitted by the Hypervisor's configured scheduling and resource-sharing policies.

3. Assessment preparation

The assessment shall have access to:

- Documentation of the Hypervisor's isolation model (CPU scheduling, memory virtualisation, IOMMU/device assignment, virtual switching, interrupt virtualisation).
- Documentation identifying which resources are shared vs dedicated across VMs.
- Documentation of the test environment, including the host platform, Hypervisor version, guest VM operating systems, and relevant network and storage configuration used for the assessment.
- A test configuration with at least two VMs on the same host, under different administrative domains or with clearly distinct identities and data.
- Ability to run test code inside each VM (e.g. to probe memory, devices, CPU load, and interrupt behaviour).
- Preconditions confirming that the Hypervisor is initialized, the VMs are deployed and running, and the relevant isolation features are enabled in accordance with the product guidance.
- The tools necessary to perform the assessment, as applicable, including tools for configuration inspection, monitoring, logging, and probing memory, device access, CPU load, or interrupt behaviour.
- Reference to relevant vendor-provided setup guides, configuration instructions, or operational manuals used to establish the assessed environment.

4. Assessment activities

The assessment shall at least:

- Memory isolation
    - Review the Hypervisor memory-assignment and mapping configuration to confirm that memory allocated to VM B is not mapped into VM A.
    - Where the product supports Hypervisor-mediated shared-memory, device-mapping, or passthrough mechanisms, attempt from VM A to use such mechanisms without authorization to obtain access to memory assigned to VM B.
    - Verify that unauthorized access is not possible and that VM A cannot read or modify memory assigned to VM B.
- I/O isolation
    - Identify virtual and, where applicable, passed-through physical devices assigned to each VM.

- From VM A, attempt to access devices or storage explicitly assigned to VM B.
- Verify that VM A can only access its own assigned devices and cannot read or modify data belonging to VM B's devices.

- CPU and interrupt isolation
  - Review documentation and configuration to confirm that CPU state and interrupts are virtualised per VM.
  - Under normal operation and during stress from VM A, verify that VM B continues to operate within the Hypervisor's configured scheduling and resource-sharing policies and does not receive spurious interrupts or exhibit behaviour indicating direct interference beyond those policies.

5. Assessment verdict

- Pass: tests and documentation confirm that CPU, memory, I/O, and interrupt-handling isolation is enforced; one VM cannot directly read or modify another VM's resources or interfere with its execution beyond the Hypervisor's configured scheduling and resource-sharing policies.
- Fail: a VM can access another VM's memory or devices, or otherwise bypass the isolation guarantees or degrade the availability of another VM beyond the Hypervisor's configured scheduling and resource-sharing policies.

6. Assessment evidence

- [SCOPE] Description of the VMs, host, and resources used in the isolation assessment.
- [CONFIG] Hypervisor configuration for CPU scheduling, memory/IOMMU, device assignment, interrupt virtualisation, and VM definitions.
- [DESIGN] Documentation of the Hypervisor isolation mechanisms across CPU, memory, I/O, and interrupts.
- [OBSERVATION] / [LOG] Results of memory/I-O/CPU/interrupt tests showing blocked access to other VMs' resources and operation within configured scheduling and resource-sharing policies under load.

6.3.1.1.2 Assessment Case AC-H-VM-ISO-002

1. Assessment reference

Requirement: REQ-H-VM-ISO-002 (Elevated)

2. Assessment objective

Verify that, where the Hypervisor exposes shared or limited resources to multiple guest virtual machines, it provides a mechanism to prevent a guest virtual machine from causing sustained denial of service to other guest virtual machines through resource exhaustion beyond the configured policy.

3. Assessment preparation

The assessment shall have access to:

- Documentation identifying the shared or limited resources exposed to multiple VMs (e.g. entropy sources, shared device backends, shared I/O services, queues, backend worker pools, or other limited shared services).
- Documentation describing the protection mechanisms used to prevent sustained denial of service through resource exhaustion (e.g. quotas, rate limiting, prioritization, admission control, or controlled degraded behaviour).
- Documentation of the relevant policy and threshold settings for the shared or limited resources under assessment.
- A test configuration with at least two VMs using one or more shared or limited resources on the same host.

- Preconditions confirming that the relevant resource-sharing functions are enabled and configured in accordance with product guidance.
- The tools necessary to generate sustained load or exhaustion attempts from one VM and to observe resource availability and service continuity for the other VM.
- Reference to relevant vendor-provided setup guides, configuration instructions, or operational manuals used to establish the assessed environment.

4. Assessment activities

The assessment shall at least:

- Identification of shared or limited resources.
- Review documentation and configuration to identify the shared or limited resources exposed to multiple VMs and the configured protection mechanisms.
- Exhaustion attempt from VM A.
- From VM A, attempt to exhaust or monopolize the identified shared or limited resource using sustained requests, load generation, or repeated service access beyond normal usage.
- Observation of effect on VM B.
- Verify whether VM B continues to obtain access to the shared or limited resource in accordance with the configured policy.
- Verify that any degradation observed is limited to the behaviour defined by the configured policy and does not result in sustained denial of service beyond that policy.
- Policy enforcement visibility.
- Verify, through configuration, status views, logs, or monitoring outputs, that the relevant protection mechanism was active and applied during the exhaustion attempt.

5. Assessment verdict

- Pass: for shared or limited resources exposed to multiple VMs, the Hypervisor implements and applies protection mechanisms that prevent sustained denial of service through resource exhaustion beyond the configured policy.
- Fail: a VM can monopolize a shared or limited resource in a way that causes sustained denial of service to other VMs beyond the configured policy, or the documented protection mechanisms are absent or ineffective.

6. Assessment evidence

- [SCOPE] Description of the shared or limited resources and VMs used in the assessment.
- [CONFIG] Configuration of quotas, rate limits, prioritization, admission control, or other relevant policy settings.
- [DESIGN] Documentation of the shared-resource protection model and enforcement mechanisms.
- [OBSERVATION] / [LOG] Test results, logs, or monitoring outputs showing behaviour of VM A and VM B during the exhaustion attempt and enforcement of the configured policy.

6.3.1.1.3 Assessment Case AC-H-VM-ISO-003

1. Assessment reference

Requirement: REQ-H-VM-ISO-003 (Advanced)

2. Assessment objective

Verify that the Hypervisor strengthens isolation between workloads sharing hardware resources by implementing mitigations against known side-channel attacks, and that these mitigations are effective in accordance with the documented security model.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing the side-channel threat model considered (e.g. cache, branch prediction, speculative execution, shared functional units, timing channels) and the corresponding mitigations implemented by the Hypervisor.
- Documentation of the test environment, including the host platform, CPU features relevant to the claimed mitigations, Hypervisor version, and the co-residency configuration of the VMs used in the assessment.
- Documentation of any configuration options associated with the claimed mitigations, where such options are provided by the product.
- A test configuration with at least two co-resident VMs on the same host.
- The tools necessary to perform side-channel-relevant measurement workloads and to observe mitigation state or configuration, where applicable.
- Reference to relevant vendor-provided setup guides, configuration instructions, or operational manuals used to establish the assessed environment.

4. Assessment activities

The assessment shall at least:

- Mitigation identification.
- Review documentation to identify the concrete mitigations provided against known side-channel attacks.
- Verify, where applicable, that any documented mitigation-related configuration settings are set in accordance with the claimed security model.
- Behaviour with the implemented mitigation set.
- Run side-channel-relevant measurement workloads in two co-resident VMs and observe whether system behaviour is consistent with the documented mitigation objectives.
- Where the product provides alternative mitigation modes or profiles, and where safely possible and consistent with product guidance, compare behaviour under different documented settings.
- Administrative or operational visibility.
- Verify, where the product provides such capability, that the implemented mitigation state or selected profile can be determined through configuration views, status commands, or logs.

5. Assessment verdict

- Pass: the Hypervisor implements mitigations against known side-channel attacks as documented, and the assessment shows behaviour consistent with strengthened isolation between workloads sharing hardware resources in accordance with the documented security model.
- Fail: no meaningful side-channel mitigations are provided for the claimed threat model, or the observed behaviour is inconsistent with the documented mitigation objectives.

6. Assessment evidence

- [CONFIG] Hypervisor configuration showing side-channel mitigation settings, where applicable.
- [DESIGN] Documentation of the side-channel threat model and implemented mitigations.

- [PLT-CAP] Platform capability evidence where relevant (e.g. CPU features used for isolation, cache/partitioning capabilities).
- [OBSERVATION] Measurement results or traces from side-channel-relevant tests demonstrating behaviour consistent with the documented mitigation objectives.

6.3.1.1.4 Assessment Case AC-H-VM-ISO-004

1. Assessment reference

Requirement: REQ-H-VM-ISO-004 (Advanced)

2. Assessment objective

Verify that all access by privileged host-environment components to guest memory, guest CPU state, or other security-relevant guest execution state is mediated through defined, auditable guest-host interfaces, and that access outside those interfaces is prevented or detected by the Hypervisor's isolation mechanisms.

3. Assessment preparation

The assessment shall have access to:

- Documentation identifying the privileged host-environment components that may access guest memory, guest CPU state, or other security-relevant guest execution state.
- Documentation of the defined guest-host interfaces through which such access is mediated, including any authorization, logging, or audit mechanisms associated with those interfaces.
- Documentation describing how unauthorized or out-of-path access is prevented or detected by the Hypervisor.
- A test environment in which at least one privileged host-environment component is present and guest-state access through the defined interface can be exercised.
- Preconditions confirming that the relevant guest-host interface, logging, and audit mechanisms are enabled in accordance with product guidance.
- The tools necessary to observe interface use, logs, audit trails, and any prevention or detection behaviour.
- Reference to relevant vendor-provided setup guides, configuration instructions, or operational manuals used to establish the assessed environment. For Type I Hypervisor implementations using a split-domain or paravirtualised architecture, the defined guest-host interfaces may be implemented through Hypervisor-mediated interdomain communication mechanisms, including grant tables, event channels, or equivalent. Documentation of these mechanisms constitutes the required interface documentation, and audit or logging of interdomain communication activity constitutes the required audit capability. Hardware-assisted memory encryption or confidential computing mechanisms are not required to satisfy REQ-H-VM-ISO-004; the requirement tests mediated interface enforcement, not hardware memory confidentiality.

4. Assessment activities

The assessment shall at least:

- Identification of mediated interfaces.
- Review documentation to identify which guest-host interfaces are defined for privileged access to guest memory, guest CPU state, or other security-relevant guest execution state.
- Use of defined interfaces.
- Exercise one or more defined guest-host interfaces from the privileged host environment in accordance with the documented administrative context.
- Verify that access occurs through the documented interface and produces the expected audit or log evidence.
- Attempted access outside defined interfaces.

- Attempt, where feasible and safe, to access guest memory, guest CPU state, or other security-relevant guest execution state from the privileged host environment outside the defined guest-host interfaces.
- Verify that such access is prevented or detected by the Hypervisor's isolation mechanisms.
- Auditability.
- Verify that access through the defined interface is auditable and that attempts outside the defined interface, where detected, generate the expected evidence or alerts.
- Where the Type I Hypervisor uses a split-domain or paravirtualised architecture, the assessment shall include review of the interdomain communication mechanism, its authorization controls, and its audit capability, and shall verify that host-environment components cannot access guest memory or CPU state other than through these defined mechanisms.

5. Assessment verdict

- Pass: privileged host-environment access to guest memory, guest CPU state, or other security-relevant guest execution state is mediated through defined, auditable guest-host interfaces, and access outside those interfaces is prevented or detected by the Hypervisor's isolation mechanisms.
- Fail: privileged host-environment components can access guest security-relevant state outside defined interfaces without prevention or detection, or the documented mediated interfaces are not auditable as required.

6. Assessment evidence

- [SCOPE] Description of the privileged host-environment components and guest-host interfaces in scope for the assessment.
- [CONFIG] Configuration of guest-host interface controls, authorization settings, and audit/logging settings.
- [DESIGN] Documentation of the guest-host mediation model and Hypervisor isolation mechanisms for privileged access.
- [OBSERVATION] / [LOG] Evidence showing authorized mediated access through defined interfaces and prevention or detection of access attempts outside those interfaces.

6.3.1.2 Assessment for Control Plane Isolation

6.3.1.2.1 Assessment Case AC-H-CP-ISO-001

1. Assessment reference
Requirement: REQ-H-CP-ISO-001 (Basic)

2. Assessment objective
Verify that:

1) The Hypervisor provides access-control mechanisms for its administrative functions and interfaces.
2) The Hypervisor can be configured so that guest virtual machines cannot directly access or invoke administrative functions/interfaces.
3) Only authenticated and authorized entities can perform management actions.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing Hypervisor administrative interfaces (CLI, GUI, APIs, management agents) and access-control mechanisms (authentication, authorization, roles).
- Documentation describing how guest VMs connect to the Hypervisor (management vs non-management channels).

- A test configuration with:
  - at least one administrative account;
  - at least one guest VM with network connectivity;
  - the ability to attempt management actions from both an admin context and from within a guest VM.

4. Assessment activities
The assessment shall at least:

1) Identify all exposed administrative interfaces and confirm from configuration that authentication and authorization controls are configured (e.g. user accounts, roles, access policies).

2) From a legitimate administrative context, perform management operations (e.g. VM create/start/stop, configuration changes) and verify that successful access requires authentication and that authorization rules (roles/permissions) are enforced.

3) From within a guest VM, attempt to:

- reach administrative endpoints (management IP/ports, APIs, management agents);
- perform management actions using unauthenticated or guest-level credentials.

Verify that:

- direct access to administrative interfaces is blocked by configuration (e.g. routing/firewall, ACLs) or requires proper administrative authentication; and
- guest VM principals are not authorized to perform management actions.

5. Assessment verdict

- Pass: access-control mechanisms exist; Hypervisor can be configured so guest VMs cannot directly invoke administrative functions; and only authenticated, authorized entities can perform management actions.
- Fail: guest VMs can directly access administrative interfaces and perform management actions; or administrative interfaces are accessible without proper authentication/authorization.

6. Assessment evidence

- [CONFIG] Access-control configuration for administrative interfaces (users, roles, policies, network ACLs).
- [DESIGN] Documentation of administrative functions/interfaces and the access-control model.
- [OBSERVATION] / [LOG] Evidence of successful authenticated/authorized admin actions and failed/blocked attempts from guest VMs.

6.3.1.2.2 Assessment Case AC-H-CP-ISO-002

1. Assessment reference
Requirement: REQ-H-CP-ISO-002 (Elevated)

2. Assessment objective
Verify that:

1) the Hypervisor enforces the use of cryptographically protected communication for network-accessible administrative interfaces, providing confidentiality and integrity protection; and

2) the Hypervisor supports the use of distinct communication channels, ports, or virtual networks to enable isolation of administrative traffic from guest workloads and guest or data-plane traffic.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing the network-accessible administrative interfaces, the supported cryptographic protection mechanisms, and the supported methods for separating administrative traffic from guest or data-plane traffic.
- Documentation describing the configuration of distinct communication channels, ports, interfaces, VLANs, virtual networks, or equivalent separation mechanisms for administrative traffic.
- A test configuration with at least one network-accessible administrative interface, at least one guest/data-plane network, and the ability to inspect protocol use, endpoint bindings, and traffic reachability.

4. Assessment activities

The assessment shall at least:

1) Configure at least one network-accessible administrative interface to use the documented cryptographically protected communication mechanism and verify, by protocol inspection or equivalent means, that confidentiality and integrity protection are in effect for that interface.
2) Configure the Hypervisor according to documentation so that administrative traffic uses distinct communication channels, ports, or virtual networks from guest workloads and guest or data-plane traffic.
3) Verify via configuration and inspection that the network-accessible administrative endpoints are bound only to the designated administrative channels, ports, or virtual networks and are not exposed on guest or data-plane networks except as explicitly configured for administration.
4) From a guest or data-plane network, attempt to connect to administrative endpoints that should be reachable only through the designated administrative channels, ports, or virtual networks and verify that such attempts are blocked or otherwise prevented by the assessed configuration.
5) From an authorized administrative host on the designated administrative path, verify that the administrative interface is reachable using the configured cryptographically protected communication mechanism.

5. Assessment verdict

- Pass: the Hypervisor enforces cryptographically protected communication for network-accessible administrative interfaces, providing confidentiality and integrity protection, and supports distinct communication channels, ports, or virtual networks that isolate administrative traffic from guest workloads and guest or data-plane traffic.
- Fail: the network-accessible administrative interfaces are not cryptographically protected as required; or confidentiality or integrity protection is absent or ineffective; or the Hypervisor does not support distinct communication channels, ports, or virtual networks to isolate administrative traffic from guest workloads and guest or data-plane traffic.

6. Assessment evidence

- [CONFIG] Configuration of cryptographic protection for network-accessible administrative interfaces and configuration of distinct communication channels, ports, interfaces, VLANs, or virtual networks used for administrative traffic.
- [DESIGN] Documentation of the administrative-interface protection model and administrative-traffic separation mechanisms.
- [OBSERVATION] / [LOG] Protocol traces, configuration views, connection tests, and logs showing encrypted administrative communication and isolation of administrative traffic from guest or data-plane traffic.

6.3.1.2.3 Assessment Case AC-H-CP-ISO-003

1. Assessment reference

Requirement: REQ-H-CP-ISO-003 (Advanced)

2. Assessment objective
Verify that:

1) The Hypervisor supports cryptographically protected communication channels for all administrative interfaces.

2) Administrative channels can be configured for mutual authentication between the Hypervisor and administrative clients/systems.

3) Use of such channels reduces the risk of unauthorized access or lateral movement (e.g. no cleartext admin protocols, no unauthenticated management channels).

3. Assessment preparation

The assessment shall have access to:

- Documentation of supported cryptographic protocols for administrative interfaces, supported cipher suites and mutual authentication options (client certificates, key-based methods).
- Documentation of procedures for configuring certificates/keys and enabling mutual authentication.
- A test configuration where administrative interfaces can be configured to use cryptographically protected, mutually authenticated channels, and where protocol-level inspection (e.g. via test tools) is possible.

4. Assessment activities
The assessment shall at least:

1) For each administrative interface type (e.g. web console, API endpoint, management agent), configure cryptographic protection according to product documentation.

2) Enable mutual authentication where supported and configure at least one administrative client with valid credentials and one with invalid or missing credentials.

3) Verify, using connection attempts and protocol inspection, that:

- administrative traffic is encrypted and uses the configured cryptographic protocol;

- servers authenticate to clients (e.g. certificate validation); and

- clients present valid credentials (keys/certificates) to complete administrative sessions for interfaces configured with mutual authentication.

4) Attempt connections without valid credentials or using obsolete/weak protocol modes (if configurable) and verify they are rejected.

5) Confirm there are no remaining administrative interfaces exposed in cleartext or without authentication in the assessed configuration.

5. Assessment verdict

- Pass: all administrative interfaces can use cryptographically protected channels; mutual authentication can be enabled and enforced; and unauthorized/unauthenticated access attempts over these channels are rejected.
- Fail: some administrative interfaces cannot be protected cryptographically; mutual authentication cannot be enabled where required; or cleartext/unauthenticated management channels remain exposed.

6. Assessment evidence

- [CONFIG] Cryptographic and authentication configuration for administrative interfaces (protocol settings, certificates/keys, mutual-auth options).
- [DESIGN] Documentation of supported protocols, mutual authentication mechanisms and security profiles for management channels.
- [LOG] / [OBSERVATION] Protocol traces and connection logs showing encrypted, mutually authenticated sessions, and failed attempts without valid credentials or with disallowed protocol modes.

6.3.1.3 Assessment for Network Plane Separation

6.3.1.3.1 Assessment Case AC-H-NP-ISO-001

1. Assessment reference
Requirement: REQ-H-NP-ISO-001 (Basic)

2. Assessment objective
Verify that:

1) The Hypervisor maintains logical segregation between management, guest and host network planes.

2) Virtual switching, routing or equivalent mechanisms ensure that traffic from one plane cannot traverse or address another plane by default.

3) Separate network contexts can be explicitly configured for each plane.

3. Assessment preparation

The assessment shall have access to:

- Documentation of the Hypervisor's network model (e.g. virtual switches, port groups, VLANs, routing, host network interfaces) and the definitions of "management", "guest" and "host" planes.
- Documentation describing how to configure separate network contexts for each plane.
- A test configuration with at least:
  - one management network/plane;
  - one guest/data network/plane used by VMs;
  - host-level services (e.g. backup/monitoring) that may represent a host network plane;
  - tools to generate and observe traffic between these networks.

4. Assessment activities
The assessment shall at least:

1) Configure the Hypervisor with distinct network contexts for management, guest and host traffic as per documentation (e.g. separate port groups, VLANs or virtual networks).

2) Verify via configuration and inspection that management interfaces are bound only to the management plane, guest VM interfaces only to the guest plane, and host services only to host/appropriate plane.

3) From each plane (management endpoint, guest VM, host service), attempt to send traffic to addresses in the other planes without adding any explicit cross-plane rules and verify that:

- by default, traffic from one plane does not reach or address endpoints in another plane;
- no routing or switching entries exist that would allow cross-plane traffic beyond what is documented as default.

5. Assessment verdict

- Pass: management, guest and host planes are logically segregated; by default, traffic from one plane cannot traverse or address another; and separate network contexts for each plane can be configured and observed.
- Fail: planes are not effectively segregated; or default configuration allows cross-plane traffic contrary to the requirement.

6. Assessment evidence

- [CONFIG] Network configuration (e.g. virtual switches, port groups, VLAN settings, IP addressing and bindings of interfaces to planes).

- [DESIGN] Documentation of network plane definitions and segregation mechanisms.
- [OBSERVATION] / [LOG] Packet captures, routing/switching tables and connection tests showing lack of cross-plane reachability in the default/segregated configuration.

6.3.1.3.2 Assessment Case AC-H-NP-ISO-002

1. Assessment reference
Requirement: REQ-H-NP-ISO-002 (Elevated)

2. Assessment objective
Verify that:

1) The Hypervisor supports configuration of explicit traffic filtering and access-control rules between network planes.
2) These rules can be enforced at the Hypervisor's virtual network layer or other enforcement points under its control.
3) Only cross-plane traffic explicitly permitted by policy (e.g. management/orchestration flows) is allowed to cross plane boundaries.

3. Assessment preparation

The assessment shall have access to:

- Documentation of mechanisms for defining traffic filtering and access-control between planes (e.g. virtual switch ACLs, distributed firewalls, router filters, security groups).
- Documentation of example policies permitting specific management/orchestration flows.
- A test configuration with:
  - management, guest and, where applicable, host planes as in AC-H-NP-ISO-001;
  - at least one management/orchestration system that legitimately needs to communicate across planes.

4. Assessment activities
The assessment shall at least:

1) Define and apply explicit cross-plane filtering rules at the Hypervisor virtual network layer or equivalent enforcement point to:
   - allow only documented management/orchestration flows (e.g. from management plane to guest VM agents on specific ports); and
   - deny all other cross-plane traffic by default.
2) From each plane, generate test traffic to other planes:
   - traffic matching permitted flows;
   - traffic that should be blocked (e.g. arbitrary guest-to-management or guest-to-host connections, unexpected ports/protocols).
3) Verify that:
   - permitted flows succeed and follow the configured policy;
   - non-permitted cross-plane traffic is blocked at the configured enforcement point; and
   - changes to the policy (e.g. adding/removing a rule) have the expected effect on cross-plane communication.

5. Assessment verdict

- Pass: the Hypervisor provides configurable filtering/access-control rules between planes; these rules are enforced at the virtual network or equivalent layer; and only explicitly permitted cross-plane traffic is allowed while other cross-plane traffic is blocked.
- Fail: cross-plane filtering cannot be configured; or enforcement is ineffective; or cross-plane traffic bypasses the configured policy.

6. Assessment evidence

- [CONFIG] Defined cross-plane filtering/ACL rules and their association with management/guest/host networks.
- [DESIGN] Documentation of enforcement mechanisms and their role in plane separation.
- [OBSERVATION] / [LOG] Connection attempts and packet traces showing success of permitted flows and blocking of non-permitted cross-plane traffic.

6.3.1.3.3 Assessment Case AC-H-NP-ISO-003

1. Assessment reference
Requirement: REQ-H-NP-ISO-003 (Advanced)

2. Assessment objective
Verify that:

1) The Hypervisor supports isolation of management traffic using either:

- dedicated physical interfaces; or
- cryptographically protected network channels.

2) Management traffic is protected against interception or modification by guest or host workloads.

3) Where dedicated physical interfaces are not used, cryptographic protection provides confidentiality, integrity and authentication equivalent to a dedicated management network.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing supported approaches for isolating management traffic (e.g. dedicated NICs/ports, out-of-band networks, IPsec/TLS tunnels, overlay management networks).
- Documentation of cryptographic protocols and authentication mechanisms used when dedicated physical interfaces are not available.
- A test configuration with:
  - either a dedicated physical management interface/network or a cryptographically protected management channel;
  - guest workloads generating network traffic;
  - the ability to capture and inspect traffic from guest-accessible paths.

4. Assessment activities
Depending on the isolation method, the assessment shall at least:

Dedicated physical interface case

1) Configure the Hypervisor so that all management interfaces use a dedicated physical NIC or physical management network segment.

2) Verify via configuration and inspection that:

- management traffic egresses/ingresses only via the dedicated physical interface/network;
- guest VMs and host non-management workloads do not have access to that interface/network (e.g. no virtual NICs or routes pointing to it).

3) From guest networks and host non-management interfaces, capture traffic and verify that management traffic is not visible on those paths.

Cryptographically protected channel case (when dedicated physical interfaces are not used)

4) Configure management traffic over a cryptographically protected channel (e.g. IPsec, TLS VPN, mutually authenticated TLS overlay) between the Hypervisor and management endpoints.

5) Verify via protocol inspection and logs that:

- management traffic is encrypted and integrity-protected;
- mutual authentication between Hypervisor and management entities is in place.

6) From guest networks and host non-management workloads, capture traffic and verify that:

- management traffic is only observable in encrypted form (no cleartext management protocols); and
- attempts to inject or modify management traffic are rejected or fail to affect the management channel.

5. Assessment verdict

- Pass: the Hypervisor supports either dedicated physical interfaces or cryptographically protected channels to isolate management traffic; management traffic is not exposed in cleartext or modifiable by guest/host workloads; and cryptographic protection, where used instead of dedicated interfaces, provides confidentiality, integrity and authentication equivalent to a dedicated management network.
- Fail: management traffic is visible or modifiable from guest or host workloads; dedicated management interfaces or equivalent cryptographic protection cannot be configured; or cleartext management flows appear on shared paths without adequate isolation.

6. Assessment evidence

- [CONFIG] Network and interface configuration showing use of dedicated management interfaces or cryptographically protected management channels.
- [DESIGN] Documentation of management traffic isolation model and, where applicable, cryptographic protocols and authentication methods.
- [PLT-CAP] Evidence of platform/network capabilities used (e.g. dedicated NICs, VPN/IPsec support).
- [OBSERVATION] Packet captures and logs demonstrating that management traffic uses the isolated interface or encrypted channel and is not exposed to guest or non-management workloads.

6.3.1.4 Assessment for Boot Integrity Verification

6.3.1.4.1 Assessment Case AC-H-B-INT-002

1. Assessment reference
Requirement: REQ-H-B-INT-002 (Elevated)

2. Assessment objective
Verify that:

1) The Hypervisor supports deployment with a boot mechanism that verifies the integrity and authenticity of its executable artefacts before those artefacts are used to initialize guest workloads or expose management interfaces.

2) The Hypervisor is delivered with, or supports the use of, verification or measurement metadata for those executable artefacts, such as digital signatures, certificates, cryptographic digests, measurements, trusted reference values, or equivalent metadata.

3) The product security documentation identifies the boot verification mechanism used or expected for the Hypervisor executable artefacts, including any Operational Environment dependency and the expected security properties of that mechanism.

4) Where the Hypervisor includes the mechanism that loads or activates its executable artefacts, this mechanism prevents loading or activation of those artefacts when verification fails.

5) Where verification status from a platform-provided boot mechanism is made available to the Hypervisor, the Hypervisor uses that status before initializing guest workloads or exposing management interfaces.

6) Where the required verification status cannot be established or is reported as failed, the Hypervisor prevents initialization of guest workloads and activation of management interfaces where such enforcement is under the control of the Hypervisor.

This assessment does not require the Hypervisor to verify firmware, platform bootloaders, Host OS components, hardware roots of trust, or other Operational Environment components, unless those components are part of the declared Hypervisor product.

3. Assessment preparation

The assessment shall have access to:

- Product security documentation identifying the Hypervisor executable artefacts covered by REQ-H-B-INT-002.
- Documentation of the boot verification mechanism used or expected for those artefacts.
- Verification or measurement metadata associated with the Hypervisor executable artefacts, such as signatures, certificates, cryptographic digests, measurements, trusted reference values, or equivalent metadata.
- Documentation of any Operational Environment dependency used for boot verification, including the expected security properties and assumptions required for the Hypervisor to be securely loaded;
- Design, configuration, or interface evidence showing how the Hypervisor uses verification status before initializing guest workloads or exposing management interfaces, where such status is made available to the Hypervisor;
- Evidence of failure behaviour where verification fails or where required verification status cannot be established, where such enforcement is under the control of the Hypervisor

4. Assessment activities

The assessment shall at least:

1) Check that the product security documentation identifies the Hypervisor executable artefacts covered by REQ-H-B-INT-002.

2) Check that the Hypervisor is delivered with, or supports the use of, verification or measurement metadata for those executable artefacts.

3) Check that the product security documentation identifies the boot verification mechanism used or expected for those executable artefacts.

4) Check whether the boot verification mechanism is provided by the Hypervisor, by another component of the declared product, or by the Operational Environment.

5) Check that any Operational Environment dependency is identified together with its expected security properties and assumptions.

6) Where the Hypervisor includes the mechanism that loads or activates its executable artefacts, check that this mechanism prevents loading or activation when verification fails.

7) Where verification status from a platform-provided boot mechanism is made available to the Hypervisor, check that the Hypervisor uses this status before initializing guest workloads or exposing management interfaces.

8) Where the required verification status cannot be established or is reported as failed, check that the Hypervisor prevents initialization of guest workloads and activation of management interfaces where such enforcement is under the control of the Hypervisor.

5. Assessment verdict

- Pass: the required evidence is provided and demonstrates that the Hypervisor supports deployment with a boot mechanism that verifies the integrity and authenticity of its executable artefacts in accordance with REQ-H-B-INT-002, including documented Operational Environment dependencies and failure behaviour where enforcement is under the control of the Hypervisor.

- Fail: the required evidence is missing or does not demonstrate that the Hypervisor supports the required boot integrity and authenticity verification capability for its executable artefacts.

The verdict shall not be Fail solely because firmware, platform bootloaders, Host OS components, hardware roots of trust, or other Operational Environment components are not verified by the Hypervisor, provided that such components are outside the declared Hypervisor product and are documented as Operational Environment dependencies.

6. Assessment evidence

- [DESIGN]:
  - Documentation identifying the Hypervisor executable artefacts covered by REQ-H-B-INT-002.
  - Documentation of the boot verification mechanism used or expected.
  - Documentation of Operational Environment dependencies, expected security properties, and assumptions.
- [CONFIG] Verification or measurement metadata associated with the covered Hypervisor executable artefacts.
- [OBSERVATION] / [TEST]:
  - Evidence showing use of verification status where such status is made available to the Hypervisor.
  - Evidence showing failure behaviour where verification fails or required verification status cannot be established, where enforcement is under the control of the Hypervisor

6.3.1.5 Assessment for Guest VM Image Integrity Verification

6.3.1.5.1 Assessment Case AC-H-IMG-INT-001

1. Assessment reference
Requirement: REQ-H-IMG-INT-001 (Basic)

2. Assessment objective
Verify that:

1) The Hypervisor implements an integrity verification mechanism for guest VM images based on trusted reference values.

2) Integrity verification is automatically invoked before executing or provisioning any guest VM.

3) If integrity verification for a guest VM image fails, the Hypervisor prevents execution or instantiation of that guest VM.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing the guest VM image integrity verification mechanism, including how trusted reference values (e.g. hashes, manifests) are established and stored.
- Documentation of the VM provisioning/startup sequence, indicating when image integrity verification is invoked.
- A test environment with:
  - at least one repository or storage location for guest VM images;
  - at least one valid guest VM image;
  - the ability to create a deliberately modified (integrity-failing) copy of a guest VM image in a controlled manner.

4. Assessment activities

The assessment shall at least:

1) Review documentation and configuration to identify:
   - which guest VM images are subject to integrity verification;
   - the trusted reference values used;
   - the point in the provisioning/startup process where verification occurs.
2) With an unmodified guest VM image, request provisioning/startup of the VM and verify, using logs or status outputs, that:
   - integrity verification is invoked automatically;
   - the VM is instantiated or started successfully when verification succeeds.
3) Replace or alter a guest VM image so that it fails integrity verification according to the documented mechanism, then request provisioning/startup of that VM. Verify that:
   - the Hypervisor reports an integrity verification failure;
   - execution or instantiation of that VM is prevented.

5. Assessment verdict

- Pass: guest VM image integrity verification based on trusted reference values is automatically invoked before VM provisioning/execution; and a verification failure for an image prevents that VM from being instantiated or executed.
- Fail: integrity verification is not implemented or not based on trusted reference values; it is not invoked automatically; or VMs can be instantiated despite integrity verification failures.

6. Assessment evidence

- [DESIGN] Documentation of the image integrity verification mechanism and its place in the VM lifecycle.
- [CONFIG] Configuration showing enabled integrity verification and reference values for guest VM images.
- [LOG] / [OBSERVATION] Records of successful verification and VM startup for a valid image, and records showing verification failure and blocked startup for a modified image.

6.3.1.5.2 Assessment Case AC-H-IMG-INT-002

1. Assessment reference

Requirement: REQ-H-IMG-INT-002 (Elevated)

2. Assessment objective
Verify that:

1) The Hypervisor verifies both integrity and authenticity of guest VM images before execution or provisioning.

2) Authenticity verification establishes that the image originates from a trusted source and has not been altered since approval.

3) The Hypervisor prevents execution or instantiation of any guest VM image whose integrity or authenticity cannot be successfully verified.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing integrity and authenticity verification for guest VM images, including trust model (e.g. signatures, certificate chains, trust stores, or equivalent cryptographic schemes).
- Documentation of how trusted publishers/sources are represented and configured.
- A test environment with:
  - one or more guest VM images created and "approved" using the trusted process (e.g. signed or otherwise bound to a trusted source);
  - images modified or created so that integrity or authenticity verification will fail (e.g. tampered content, invalid or missing signature, signature under an untrusted key), prepared in a controlled way.

4. Assessment activities
The assessment shall at least:

1) Review documentation and configuration to identify:

- how integrity and authenticity checks are combined;
- what constitutes a "trusted source" and how trust anchors or approval data are configured.

2) For an image that is correctly produced and approved:

- request provisioning/startup of a VM from that image;
- verify that integrity and authenticity verification succeed;
- confirm that the VM is instantiated/executed.

3) For images that are invalid in different ways (e.g. modified after approval, missing or invalid signature, signed by an untrusted source):

- request provisioning/startup;
- verify that the Hypervisor reports integrity/authenticity verification failure or inability to verify;
- confirm that execution/instantiation of such VMs is prevented.

5. Assessment verdict

- Pass: integrity and authenticity of guest VM images are verified prior to provisioning/execution; only images from trusted sources that have not been altered since approval can be instantiated; and images whose integrity/authenticity cannot be verified are blocked.
- Fail: authenticity is not verified; untrusted or modified images can be instantiated; or failure to verify integrity/authenticity does not prevent VM instantiation.

6. Assessment evidence

- [DESIGN] Documentation of the guest VM image trust model, including integrity and authenticity verification and trusted sources.
- [CONFIG] Trust store or equivalent configuration defining trusted publishers/sources and related policies.
- [LOG] / [OBSERVATION] Records showing successful verification and startup for approved images and failure/blocked startup for images with invalid integrity or authenticity.

6.3.1.5.3 Assessment Case AC-H-IMG-INT-003

1. Assessment reference
Requirement: REQ-H-IMG-INT-003 (Advanced)

2. Assessment objective
Verify that:

1) The Hypervisor uses integrity and authenticity verification for all guest VM images before execution or provisioning, based on trust anchors stored as protected trust material.

2) Protected trust material (keys, certificates or other trust anchors) is stored and managed so that it is not modifiable during normal operational state via standard administrative or guest-facing interfaces.

3) If verification of a guest VM image using the protected trust material fails, or the verification status cannot be established, the Hypervisor prevents execution or instantiation of that guest VM.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing:
  - the nature and storage of protected trust material (e.g. trust stores, key vaults, secure configuration objects);
  - the protections that prevent modification of trust material during normal operation;
  - the process for controlled update of trust material (e.g. dedicated security management procedure).
- A test environment with:
  - at least one valid guest VM image verified against the protected trust material;
  - the ability to simulate conditions where verification fails or the trust anchors no longer match the image (in a controlled manner);
  - an administrative interface reflective of "normal operational state".

4. Assessment activities
The assessment shall at least:

1) Review documentation and configuration to confirm:
   - which trust anchors are used to verify guest VM images;
   - where and how this trust material is stored;
   - which operations are permitted in normal operational state versus dedicated security update procedures.
2) In normal operation, attempt to modify or replace trust material through standard administrative channels (e.g. routine configuration interfaces) and verify that such modifications are prevented or require a controlled security management process.
3) With valid trust material and a correctly approved guest VM image, request provisioning/startup and verify that integrity/authenticity verification succeeds and the VM is instantiated.

4) Create a test condition where verification fails or becomes indeterminate for a guest VM image when checked against the protected trust material (e.g. image altered after approval, trust anchor no longer matches image), and request provisioning/startup. Verify that:

- verification failure or inability to establish verification status is detected;
- the Hypervisor prevents execution or instantiation of that VM.

5. Assessment verdict

- Pass: all guest VM images are verified for integrity and authenticity against protected trust material; trust anchors are not modifiable during normal operation; and images that fail or cannot be verified against this material are prevented from execution/instantiation.
- Fail: trust material is modifiable in normal operation without a controlled process; images can be instantiated without verification against protected trust anchors; or failed/indeterminate verification does not prevent VM instantiation.

6. Assessment evidence

- [DESIGN] Documentation of protected trust material, its storage and update procedures, and its use in image verification.
- [CONFIG] Configuration of trust stores or equivalent structures, including permissions or controls that protect them in normal operation.
- [LOG] / [OBSERVATION] Evidence of successful verification and startup for valid images, and evidence of blocked startup when verification against protected trust material fails or is not possible.

6.3.1.6 Assessment for Runtime Integrity Protection

6.3.1.6.1 Assessment Case AC-H-RP-INT-002

1. Assessment reference
Requirement: REQ-H-RP-INT-002 (Elevated)

2. Assessment objective
Verify that:

1) the Hypervisor supports integrity protection for its security-critical runtime code, configuration state, and control logic used to enforce security policies or isolation;
2) the Hypervisor supports mechanisms ensuring that security-critical configuration state and control logic can be modified only through authenticated and authorized interfaces or processes;
3) where runtime integrity protection relies on mechanisms provided by the Operational Environment, the Operational Environment dependency is identified in the product security documentation.

3. Assessment preparation

The assessment shall have access to:

- product security documentation identifying the security-critical runtime code, configuration state, and control logic covered by REQ-H-RP-INT-002;
- design or configuration documentation describing the mechanisms used to protect the integrity of the covered runtime code, configuration state, and control logic;
- documentation identifying the interfaces or processes through which security-critical configuration state and control logic can be modified;
- documentation showing that such modification interfaces or processes require authentication and authorization;

- documentation of any Operational Environment dependency used for runtime integrity protection, including expected security properties.

4. Assessment activities
The assessment shall at least:

1) Check that the product security documentation identifies the security-critical runtime code, configuration state, and control logic covered by REQ-H-RP-INT-002.

2) Check that the Hypervisor supports mechanisms to protect the integrity of the identified security-critical runtime code, configuration state, and control logic.

3) Check that security-critical configuration state and control logic can be modified only through identified interfaces or processes.

4) Check that the identified modification interfaces or processes require authentication.

5) Check that the identified modification interfaces or processes require authorization.

6) Where runtime integrity protection relies on the Operational Environment, check that the dependency and expected security properties are identified in the product security documentation.

5. Assessment verdict

- Pass: the required evidence demonstrates that the Hypervisor supports integrity protection for its security-critical runtime code, configuration state, and control logic, and that security-critical configuration state and control logic can be modified only through authenticated and authorized interfaces or processes.
- Fail: the required evidence is missing or does not demonstrate that the Hypervisor supports the required runtime integrity protection mechanisms or authenticated and authorized modification paths.

6. Assessment evidence

- [DESIGN]:
  - Documentation identifying the covered security-critical runtime code, configuration state, and control logic.
  - Description of the runtime integrity protection mechanisms.
  - Operational Environment dependency documentation, where applicable.
- [CONFIG] Documentation of authenticated and authorized modification interfaces or processes.

6.3.1.7 Assessment for Remote Attestation

6.3.1.7.1 Assessment Case AC-H-RA-INT-003

1. Assessment reference
Requirement: REQ-H-RA-INT-003 (Advanced, optional capability)

2. Assessment objective
Verify that, where remote attestation is included in the declared product configuration:

1) the Hypervisor provides generation of attestation evidence describing boot-time or launch-time measurements of its security-critical executable artefacts;

2) the attestation evidence is cryptographically verifiable by an authorized verifier;

3) the attestation evidence is bound to the Hypervisor instance for which the measurements were produced;

4) the Hypervisor protects attestation keys and measurement data against unauthorized access, disclosure, or replay;

5) the attestation evidence is limited to the minimum information necessary to verify the integrity of the measured Hypervisor executable artefacts;

6) where attestation evidence relies on platform, Host OS, firmware, hardware root of trust, or external attestation services outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms are identified as Operational Environment dependencies in the product security documentation;

7) where configuration-state attestation is claimed in the product security documentation, the configuration-state measurements are subject to the same cryptographic verifiability and key protection obligations as the boot-time or launch-time measurements required by REQ-H-RA-INT-003.

This assessment does not require attestation of Hypervisor runtime policy state, arbitrary Hypervisor memory, guest memory, guest device state, transient scheduling state, workload-dependent runtime state, or other dynamic runtime data.

This assessment does not require configuration-state attestation unless configuration-state attestation is claimed in the product security documentation.

3. Assessment preparation
The assessment shall have access to:

- Product security documentation identifying whether remote attestation is included in the declared product configuration.
- Documentation identifying the Hypervisor executable artefacts covered by attestation evidence.
- Documentation of the boot-time or launch-time measurements generated or exposed by the Hypervisor.
- Representative attestation evidence or attestation evidence format.
- Documentation showing how the attestation evidence is cryptographically verifiable by an authorized verifier.
- Documentation showing how the attestation evidence is bound to the Hypervisor instance for which the measurements were produced.
- Documentation of mechanisms protecting attestation keys and measurement data against unauthorized access, disclosure, or replay.
- Documentation identifying any Operational Environment dependency used for attestation, including expected security properties.
- Documentation identifying whether configuration-state attestation is claimed, and where claimed, documentation showing that configuration-state measurements are subject to the same cryptographic verifiability and key protection obligations as the boot-time or launch-time measurements.

4. Assessment activities
The assessment shall at least:

1) Check whether remote attestation is included in the declared product configuration.

2) If remote attestation is not included in the declared product configuration, assign Not Applicable for this assessment case.

3) Check that the attestation evidence describes boot-time or launch-time measurements of security-critical Hypervisor executable artefacts.

4) Check that the attestation evidence is cryptographically verifiable by an authorized verifier.

5) Check that the attestation evidence is bound to the Hypervisor instance for which the measurements were produced.

6) Check that attestation keys and measurement data are protected against unauthorized access, disclosure, or replay.

7) Check that the attestation evidence is limited to the minimum information necessary to verify the integrity of the measured Hypervisor executable artefacts.

8) Check that any platform, Host OS, firmware, hardware root of trust, or external attestation service relied upon for attestation is identified as an Operational Environment dependency, including expected security properties.

9) Check whether configuration-state attestation is claimed in the product security documentation.

10) Where configuration-state attestation is claimed, check that configuration-state measurements are subject to the same cryptographic verifiability and key protection obligations as the boot-time or launch-time measurements required by REQ-H-RA-INT-003.

11) Check that the assessment does not require attestation of Hypervisor runtime policy state, arbitrary Hypervisor memory, guest memory, guest device state, transient scheduling state, workload-dependent runtime state, or other dynamic runtime data.

5. Assessment verdict

- Pass: remote attestation is included in the declared product configuration and the required evidence demonstrates that the Hypervisor generates cryptographically verifiable attestation evidence for boot-time or launch-time measurements of security-critical executable artefacts in accordance with REQ-H-RA-INT-003, and, where configuration-state attestation is claimed, the configuration-state measurements are subject to the same cryptographic verifiability and key protection obligations as the boot-time or launch-time measurements.
- Fail: remote attestation is included in the declared product configuration, and the required evidence is missing or does not demonstrate compliance with REQ-H-RA-INT-003.
- Not Applicable: remote attestation is not included in the declared product configuration.

6. Assessment evidence

- [DESIGN]:
  - Documentation identifying the Hypervisor executable artefacts covered by attestation evidence.
  - Description of boot-time or launch-time measurements.
  - Cryptographic verifiability and instance-binding description.
  - Attestation key and measurement data protection evidence.
  - Operational Environment dependency documentation, where applicable.
- [CONFIG] Representative attestation evidence or attestation evidence format.

6.3.1.7.2 Assessment Case AC-H-RA-INT-004

1. Assessment reference
Requirement: REQ-H-RA-INT-004 (Advanced, optional capability)

2. Assessment objective
Verify, where the Hypervisor claims support for privacy-preserving attestation options and where attestation evidence is intended to be used in remote verification scenarios, that:

1) The Hypervisor supports configuration of privacy-preserving attestation options for remote attestation use cases that reduce linkability between attestation events.

2) These privacy-preserving options preserve the ability of authorized verifiers to assess the integrity and configuration state of the Hypervisor.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing:
  - the Hypervisor remote attestation model, including how Hypervisor identity and instance information are represented in attestation evidence;
  - the privacy-preserving options available (for example use of pseudonymous identifiers, separation of identity and integrity information, control of attestation frequency or granularity);
  - how these options are configured;
  - the expected impact of each option on attestation evidence fields and verifier behaviour.
- A test configuration with:
  - a Hypervisor deployment where remote attestation to one or more authorized verifiers is enabled;
  - the ability to configure and modify the documented privacy-preserving options;
  - tools or procedures to trigger multiple attestation events and capture the resulting attestation evidence under different privacy configurations;
  - access to an authorized verifier or verification tool capable of validating integrity and configuration state from the attestation evidence in both baseline and privacy-enhanced modes.

4. Assessment activities
The assessment shall at least:

1) Review documentation and Hypervisor settings to identify:
   - the default behaviour of remote attestation when privacy-preserving options are not enabled;
   - the specific privacy options available and their documented effects on identifiers, metadata, and attestation frequency or granularity;
   - any prerequisites or constraints for enabling these options.
2) With privacy-preserving options disabled or in their default state:
   - trigger multiple attestation events over time;
   - collect the corresponding attestation evidence;
   - Analyse the evidence to determine the stability and linkability of identifiers and other potentially identifying fields across events;
   - use the authorized verifier or tool to confirm that the integrity and configuration state of the Hypervisor can be correctly assessed from this baseline evidence set.
3) With one or more privacy-preserving options enabled as documented:
   - trigger multiple attestation events under the privacy-enhanced configuration;
   - collect the resulting attestation evidence;
   - compare this evidence set with the baseline to verify that linkability between attestation events is reduced in the documented manner (for example through pseudonymous or rotated identifiers, separation of identity and integrity data, or reducedauthorized identifying metadata);
   - verify, using the authorized verifier or tool, that the integrity and configuration state of the Hypervisor can still be reliably assessed using the privacy-enhanced evidence and any associated metadata or policies.

5. Assessment verdict

- Pass: when remote attestation is used, the Hypervisor provides configurable privacy-preserving attestation options that, when enabled, reduce linkability between attestation events as documented and still allow authorized verifiers to correctly assess the Hypervisor's integrity and configuration state.
- Fail: privacy-preserving options do not meaningfully reduce linkability between attestation events; or enabling these options prevents authorized verifiers from correctly assessing the Hypervisor's integrity and configuration state.

6. Assessment evidence

- [DESIGN] Documentation of the remote attestation model, identity representation, and privacy-preserving options, including their expected effects on attestation evidence and verifier behaviour.
- [CONFIG] Hypervisor and verifier configuration showing remote attestation settings and selected privacy-preserving options.
- [LOG] Records of attestation events and related configuration changes, including timestamps and any metadata relevant to privacy options.
- [OBSERVATION] Sets of attestation evidence collected before and after enabling privacy-preserving options, with verifier outputs showing successful integrity and configuration assessment in both modes and reduced linkability when privacy options are enabled.

6.3.1.8 Assessment for Administrative Authentication

6.3.1.8.1 Assessment Case AC-H-ADMIN-AUTH-001

1. Assessment reference
Requirement: REQ-H-ADMIN-AUTH-001 (Basic)

2. Assessment objective
Verify that:

1) all interfaces that provide administrative functionality require authentication before any administrative operation is performed.
2) administrative accounts use unique, non-default credentials; and
3) where password-based authentication is supported, the Hypervisor enforces password complexity requirements and a mechanism limiting repeated failed authentication attempts, with configurable parameters.

3. Assessment preparation
The assessment shall have access to:

- The declared product composition and list of components exposing administrative interfaces.
- A list of all administrative interfaces (e.g. local CLI, remote management, APIs, VM management consoles).
- Documentation on administrative accounts, password policy options, and the mechanism limiting repeated failed authentication attempts, including its configurable parameters.
- At least one initialized configuration with one or more administrative accounts and access to the identified interfaces.

4. Assessment activities
The assessment shall at least:

1) Confirm from documentation and configuration which interfaces provide administrative functionality and verify that, in the assessed configuration, each such interface requires authentication before administrative operations.

2) Review administrative account configuration and verify that administrative accounts use unique, non-default credentials in the assessed configuration.

3) Where password-based authentication is supported, confirm that password complexity requirements can be configured and verify that non-compliant passwords are not accepted for administrative accounts in the assessed configuration.

4) Where password-based authentication is supported, confirm that the mechanism limiting repeated failed authentication attempts can be configured and verify that repeated failed authentication attempts are limited in accordance with the configured parameters.

5. Assessment verdict

- Pass: all administrative interfaces require authentication; administrative accounts use unique, non-default credentials; and, where password-based authentication is supported, password complexity requirements and a mechanism limiting repeated failed authentication attempts are configurable and enforced.
- Fail: at least one administrative interface allows unauthenticated administrative operations; or administrative accounts rely on non-unique or default credentials; or, where password-based authentication is supported, password complexity requirements or the mechanism limiting repeated failed authentication attempts are not configurable or not enforced.

6. Assessment evidence

- [SCOPE] Declared product composition and list of administrative interfaces considered.
- [CONFIG] Administrative authentication configuration, password-policy configuration, and configuration of the mechanism limiting repeated failed authentication attempts.
- [DESIGN] Description of the administrative access model, credential model, password-policy mechanism, and failed-attempt limitation mechanism.
- [LOG] / [OBSERVATION] Logs, screenshots, or console outputs showing authentication enforcement, rejection of non-compliant passwords, and enforcement of repeated failed-attempt limitation.

6.3.1.8.2 Assessment Case AC-H-ADMIN-AUTH-002

1. Assessment reference
Requirement: REQ-H-ADMIN-AUTH-002 (Elevated)

2. Assessment objective
Verify that:

1) Administrative interfaces require authenticated access and support cryptographic key-based authentication for administrative accounts.

2) Password-based login for administrative interfaces can be disabled.

3) Algorithms and key lengths for key-based administrative authentication conform to the cryptographic profile declared by the manufacturer, aligned with Annex K.

4) Private keys used for administrative authentication are protected from access or reuse by guest workloads and other unprivileged components.

3. Assessment preparation
The assessment shall have access to:

- Documentation on key-based administrative authentication for each administrative interface.
- Documentation of applicable cryptographic requirements or profiles for administrative authentication.
- Documentation of key storage locations and access-control mechanisms for administrative private keys.
- At least one configuration where administrative accounts use key-based authentication on relevant interfaces and password-based login can be disabled.

4. Assessment activities
The assessment shall at least:

1) Confirm from documentation and configuration that administrative accounts can use cryptographic key-based authentication on relevant interfaces and verify that an administrative account can obtain administrative access using the key-based mechanism.

2) Confirm that the product provides configuration to disable password-based login for administrative interfaces and verify that, where disabled, password-only administrative logins are rejected while key-based authentication remains available.

3) Review configuration or observable protocol characteristics to confirm that algorithms, key lengths and protocol versions used for key-based administrative authentication comply with the declared cryptographic profile and that weaker parameters identified as insufficient are not used.

4) Review the design and configuration of key storage for administrative private keys and confirm that guest workloads and other unprivileged components cannot read, export or otherwise misuse these keys.

5. Assessment verdict

- Pass: administrative interfaces require authenticated access and support key-based administrative authentication; password-based login for administrative interfaces can be disabled and is effectively blocked where configured; cryptographic parameters comply with the declared profile; and administrative private keys are protected from guest and unprivileged access.

- Fail: key-based administrative authentication is not available or not functional; or password-based login cannot be disabled or remains effective when disabled; or cryptographic parameters are weaker than the declared profile; or administrative private keys are accessible to guest workloads or other unprivileged components.

6. Assessment evidence

- [CONFIG] Configuration for key-based administrative authentication, password-login disablement and cryptographic settings.

- [DESIGN] Documentation of key management and key storage architecture and cryptographic profile for administrative authentication.

- [LOG] / [OBSERVATION] Evidence of successful key-based administrative authentication and rejection of password-only attempts when disabled.

- [PLT-CAP] Where applicable, evidence of platform capabilities used for key protection.

6.3.1.8.3 Assessment Case AC-H-ADMIN-AUTH-003

1. Assessment reference
Requirement: REQ-H-ADMIN-AUTH-003 (Advanced)

2. Assessment objective
Verify that:

1) Certificate-based strong authentication is supported and enforceable for administrative accounts on administrative interfaces declared in scope.

2) Certificates used for administrative authentication are validated against a trusted PKI, including chain of trust, validity period and, where supported, revocation status.

3) Administrative accounts configured for certificate-based strong authentication cannot use weaker authentication methods, except for explicitly documented emergency mechanisms.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing certificate-based administrative authentication for each relevant interface (e.g. mutual TLS, client certificates).

- Documentation of PKI integration: trusted CAs, account/certificate mapping, revocation mechanisms (CRL, OCSP or equivalent).
- At least one configuration with administrative accounts using certificates issued by a trusted CA, and untrusted, expired and, where supported, revoked certificates (or equivalent) for negative checks.

4. Assessment activities
The assessment shall at least:

1) Confirm from documentation and configuration that administrative accounts can be configured for certificate-based authentication on administrative interfaces declared in scope and verify that administrative access is granted only when a valid certificate is presented in line with the configured policy.

2) Confirm that the product validates administrative certificates against the configured trusted CAs and checks validity period, and verify that untrusted, expired or not-yet-valid certificates are rejected; where revocation is supported, verify that revoked certificates are rejected when revocation information is available.

3) Review configuration options for enforcing certificate-based strong authentication and verify that administrative accounts configured for such authentication cannot obtain administrative access using weaker methods alone (e.g. password-only), except where an explicitly documented emergency mechanism applies and its conditions are met.

5. Assessment verdict

- Pass: certificate-based authentication is available and enforceable for administrative accounts on administrative interfaces declared in scope; PKI validation (trusted CAs, validity, and, where supported, revocation) behaves as documented; downgrade to weaker authentication for accounts configured for certificate-based strong authentication is not permitted except under documented emergency mechanisms.
- Fail: certificate-based administrative authentication cannot be configured or is not functioning; or PKI validation is incomplete or inconsistent with documentation; or downgrade to weaker methods is possible for accounts intended to use certificate-based strong authentication.

6. Assessment evidence

- [CONFIG] Configuration for certificate-based administrative authentication, trusted CA configuration, revocation settings and account-to-certificate mappings.
- [DESIGN] Documentation of the architecture for certificate-based administrative authentication.
- [LOG] / [OBSERVATION] Evidence of successful authentication with valid administrative certificates and rejection of untrusted, expired or revoked certificates, and of rejected downgrade attempts.
- [PLT-CAP] Where relevant, evidence of platform capabilities used to support trust anchors or protect key and trust stores.

6.3.1.9 Assessment for Service Authentication

6.3.1.9.1 Assessment Case AC-H-SERV-AUTH-001

1. Assessment reference
Requirement: REQ-H-SERV-AUTH-001 (Basic)

2. Assessment objective
Verify that:

1) External services accessing Hypervisor management functions or data are authenticated.

2) Each service account uses distinct, non-default credentials and no shared or hard-coded service credentials are relied upon.

3) A configurable lockout mechanism exists and is enforced for service accounts after consecutive failed authentication attempts.

3. Assessment preparation

The assessment shall have access to:

- Declared product composition and list of external services/interfaces accessing management functions or data (e.g. orchestration, monitoring, backup).
- Documentation on service accounts, default/built-in service credentials, credential policy and lockout options.
- At least one configuration with representative external services integrated via service accounts.

4. Assessment activities
The assessment shall at least:

1) Identify all external services accessing management functions or data and verify that each is required to authenticate before access.

2) Review documentation/configuration to confirm distinct credentials per service account and absence of shared or hard-coded service credentials in normal operation.

3) Confirm that lockout thresholds for service accounts can be configured and verify that repeated failed authentication attempts trigger lockout according to the configured parameters.

5. Assessment verdict

- Pass: external services accessing management functions or data require authentication; each service account uses distinct, non-default credentials without reliance on shared/hard-coded credentials; and a configurable lockout mechanism for service accounts exists and is enforced.
- Fail: unauthenticated service access is possible; or shared/default/hard-coded service credentials are required for normal use; or a lockout mechanism for service accounts is missing, not configurable, or not enforced.

6. Assessment evidence

- [SCOPE] List of external services and interfaces accessing management functions or data.
- [CONFIG] Service authentication, service account and lockout configuration.
- [DESIGN] Description of service account model, handling of default/built-in service credentials and lockout mechanism.
- [LOG] / [OBSERVATION] Evidence of enforced service authentication and lockout after failed attempts.

6.3.1.9.2 Assessment Case AC-H-SERV-AUTH-002

1. Assessment reference
Requirement: REQ-H-SERV-AUTH-002 (Elevated)

2. Assessment objective
Verify that:

1) External services accessing management functions or data require authenticated access and can use certificate-based mutual authentication.

2) Algorithms and key lengths for service mutual authentication conform to the declared cryptographic profile, aligned with Annex K.

3) Private keys used for service authentication are protected from extraction or reuse by guest workloads and other unprivileged components.

3. Assessment preparation

The assessment shall have access to:

- Documentation on certificate-based mutual authentication for external services.
- Documentation on cryptographic profiles for service authentication (algorithms, key sizes, disallowed options).
- Documentation on key management and storage for service authentication keys, including separation from guest workloads/unprivileged components.
- At least one configuration where representative services access management functions or data using certificate-based mutual authentication.

4. Assessment activities
The assessment shall at least:

1) Confirm from documentation/configuration that services accessing management functions or data are authenticated and that certificate-based mutual authentication is supported; verify that a service configured for mutual authentication can establish an authenticated session and access management functions or data.

2) Review configuration/protocol characteristics to confirm that algorithms, key lengths and protocol versions used for service mutual authentication comply with the declared cryptographic profile and that weaker/non-compliant options are not used.

3) Review key storage design and configuration to confirm that private keys used for service authentication reside in protected key stores and cannot be read, exported or misused by guest workloads or other unprivileged components.

5. Assessment verdict

- Pass: authenticated access and certificate-based mutual authentication are available for services accessing management functions or data; cryptographic parameters conform to the declared profile; and service authentication keys are protected from guest and unprivileged access.
- Fail: service access does not consistently require authentication; or mutual authentication is not available/functional where claimed; or cryptographic parameters are weaker than the declared profile; or service authentication keys are accessible to guest workloads or other unprivileged components.

6. Assessment evidence

- [CONFIG] Mutual authentication configuration for services, including certificates and crypto settings.
- [DESIGN] Service authentication and key management architecture, including storage and protection of service keys and applied cryptographic profile.
- [LOG] / [OBSERVATION] Evidence of successful certificate-based mutual authentication and rejection of non-compliant or unauthenticated connections.
- [PLT-CAP] Where relevant, evidence of platform capabilities used to protect service authentication keys.

6.3.1.9.3 Assessment Case AC-H-SERV-AUTH-003

1. Assessment reference
Requirement: REQ-H-SERV-AUTH-003 (Advanced)

2. Assessment objective
Verify that:

1) Certificate-based mutual authentication is enforced for service interactions that access Hypervisor management functions or data.

2) Certificate paths for service authentication are validated against a trusted PKI.

3) Certificates used for service authentication are bound to specific service identities.

4) Certificates are checked for validity, expiration and revocation status before access is granted.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing enforcement of certificate-based mutual authentication for service interactions accessing management functions or data, including in-scope interfaces.
- Documentation of the PKI trust model (trusted CAs, trust stores, path validation rules) used for service authentication.
- Documentation on how service identities are bound to certificates and how validity and revocation information (CRL, OCSP or equivalent) is obtained and processed.
- At least one configuration in which all such service interactions use certificate-based mutual authentication, with valid and invalid/expired/revoked service certificates (or equivalent conditions) available for testing.

4. Assessment activities
The assessment shall at least:

1) Confirm from documentation/configuration that certificate-based mutual authentication is enforced for service interactions accessing management functions or data and verify that such interactions cannot be established without successful mutual certificate authentication.
2) Review identity-binding rules and verify that certificates used for service authentication are associated with specific service identities as documented.
3) Confirm that certificate paths are validated against a trusted PKI and verify that certificates from untrusted CAs or with invalid chains are rejected.
4) Verify that the Hypervisor checks certificate validity period and rejects expired or not-yet-valid certificates, and, where revocation mechanisms are supported, that revoked (or otherwise invalidated) service certificates are rejected before access is granted.

5. Assessment verdict

- Pass: certificate-based mutual authentication is enforced for service interactions accessing management functions or data; certificate paths are validated against a trusted PKI; certificates are bound to specific service identities; and validity, expiration and revocation (or equivalent) are checked and enforced before access is granted.
- Fail: service interactions can occur without enforced certificate-based mutual authentication; or certificate paths are not properly validated; or certificates are not effectively bound to specific service identities; or validity/expiration/revocation checks are missing or ineffective.

6. Assessment evidence

- [CONFIG] Configuration enforcing certificate-based mutual authentication for services, including trust stores, CA configuration and identity bindings.
- [DESIGN] Documentation of the PKI trust model for service authentication, identity binding and revocation handling.
- [LOG] / [OBSERVATION] Evidence of successful service interactions with valid certificates and rejection of unauthenticated, untrusted, expired or revoked certificates.
- [PLT-CAP] Where relevant, evidence of platform capabilities used to support certificate validation and PKI trust.

6.3.1.10 Assessment for Administrative Authorization

6.3.1.10.1 Assessment Case AC-H-ADMIN-AUTHZ-001

1. Assessment reference
Requirement: REQ-H-ADMIN-AUTHZ-001 (Basic)

2. Assessment objective
Verify that:

1) All administrative actions are restricted to authenticated administrative accounts.

2) A default-deny policy is enforced for access to management functions.

3. Assessment preparation

The assessment shall have access to:

- Documentation of administrative roles, accounts and management interfaces.
- Documentation of the authorization model for administrative actions (including default behaviour).
- At least one configuration with multiple administrative accounts and access to management functions.

4. Assessment activities
The assessment shall at least:

1) Verify that unauthenticated users and non-administrative accounts cannot perform administrative actions on any management interface.

2) Confirm from configuration and tests that unspecified administrative actions are denied by default, and that only explicitly authorized administrative actions are permitted for authenticated accounts.

5. Assessment verdict

- Pass: administrative actions can only be performed by authenticated administrative accounts and any non-authorized administrative action is denied by default.
- Fail: unauthenticated or non-administrative users can perform administrative actions; or unspecified administrative actions are allowed.

6. Assessment evidence

- [CONFIG] Administrative authorization configuration and default policies.
- [DESIGN] Description of administrative authorization model and default-deny behaviour.
- [LOG] / [OBSERVATION] Evidence of denied attempts by unauthenticated/non-admin users and permitted actions only for authorized administrative accounts.

6.3.1.10.2 Assessment Case AC-H-ADMIN-AUTHZ-002

1. Assessment reference
Requirement: REQ-H-ADMIN-AUTHZ-002 (Elevated)

2. Assessment objective
Verify that the Hypervisor enforces granular authorization for administrative users and administrative operations, binding permitted actions to authenticated identities and enforcing least-privilege permissions by default. The assessment shall not require a specific authorization model, provided that the implemented model supports granular permissions and least-privilege enforcement. The implemented authorization model may be role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing the implemented Hypervisor administrative authorization model, including whether it is role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.
- Documentation identifying administrative operations protected by the authorization model.
- Documentation identifying authenticated administrative identities, permissions, policy rules, or equivalent authorization bindings.

- A test configuration with at least two authenticated administrative identities or accounts having different permissions for representative administrative operations.
- Tools or interfaces to attempt permitted and non-permitted administrative operations using those authenticated identities.

4. Assessment activities
The assessment shall at least:

1) Identify the Hypervisor administrative operations covered by REQ-H-ADMIN-AUTHZ-002.
2) Review the implemented authorization model and confirm that permitted administrative actions are bound to authenticated identities.
3) Verify that least-privilege permissions are enforced by default for administrative identities or accounts.
4) Attempt a permitted administrative action using an authenticated identity authorized for that action and verify that the action succeeds.
5) Attempt the same administrative action using an authenticated identity not authorized for that action and verify that the action is denied.
6) Verify that the evidence demonstrates granular authorization without requiring a specific authorization model.

5. Assessment verdict

- Pass: the Hypervisor enforces granular authorization for administrative operations, binds permitted administrative actions to authenticated identities, and enforces least-privilege permissions by default, regardless of whether the implemented model is role-based, attribute-based, policy-based, capability-based, or equivalent.
- Fail: the Hypervisor does not enforce granular authorization for administrative operations; or permitted administrative actions are not bound to authenticated identities; or least-privilege permissions are not enforced by default; or an authenticated identity can perform an administrative action for which it is not authorized.

6. Assessment evidence

- [DESIGN] Documentation of the implemented Hypervisor administrative authorization model and its binding of permissions to authenticated identities.
- [CONFIG] Authorization policy, role, attribute, rule, capability, or equivalent configuration showing permitted and non-permitted administrative actions.
- [LOG] / [OBSERVATION] Evidence of permitted administrative actions succeeding for authorized identities and non-permitted administrative actions being denied for unauthorized identities.

6.3.1.10.3 Assessment Case AC-H-ADMIN-AUTHZ-003

1. Assessment reference
Requirement: REQ-H-ADMIN-AUTHZ-003 (Advanced)

2. Assessment objective
Verify that the Hypervisor enforces fine-grained, policy-driven, or context-aware authorization for administrative operations, including separation of duties, contextual policy evaluation, or time-bound privilege elevation where applicable to the corresponding requirement and to the declared product configuration. Where time-bound or just-in-time privilege elevation is not used by the declared product configuration, the assessment shall not require it.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing the Hypervisor fine-grained, policy-driven, or context-aware authorization model.

- Documentation identifying separation-of-duties rules, contextual policy inputs, time-bound privilege elevation, just-in-time privilege elevation, or equivalent mechanisms supported by the Hypervisor, where applicable.
- Documentation identifying which of these mechanisms are part of the declared product configuration.
- A test configuration with representative administrative identities, permissions, policy inputs, contextual attributes, or privilege-elevation workflows, as applicable to the declared product configuration.

4. Assessment activities
The assessment shall at least:

1) Review the documented fine-grained, policy-driven, or context-aware administrative authorization model.
2) Verify that the Hypervisor enforces the applicable authorization policy for representative administrative operations.
3) Where separation of duties is supported or required by the declared configuration, verify that conflicting or incompatible administrative operations are not permitted to the same identity unless explicitly authorized by policy.
4) Where context-aware enforcement is supported or applicable, verify that authorization decisions take the documented contextual inputs into account.
5) Where time-bound or just-in-time privilege elevation is part of the declared product configuration, verify that elevated privileges are granted only according to the documented policy and are removed or expire according to that policy.

5. Assessment verdict

- Pass: the Hypervisor enforces fine-grained, policy-driven, or context-aware authorization for administrative operations, including separation of duties, contextual policy evaluation, or time-bound privilege elevation where applicable to the requirement and declared product configuration.
- Fail: the Hypervisor does not enforce fine-grained, policy-driven, or context-aware administrative authorization where required; or separation-of-duties, contextual policy evaluation, or time-bound privilege elevation is claimed but not implemented or not effective in the declared product configuration.

6. Assessment evidence

- [DESIGN] Documentation of the fine-grained, policy-driven, or context-aware administrative authorization model.
- [CONFIG] Authorization policy, contextual policy, separation-of-duties configuration, privilege-elevation configuration, or equivalent configuration, as applicable.
- [LOG] / [OBSERVATION] Evidence that administrative authorization decisions are enforced according to the documented policy and declared product configuration.

6.3.1.11 Assessment for Service Authorization

6.3.1.11.1 Assessment Case AC-H-SERV-AUTHZ-001

1. Assessment reference
Requirement: REQ-H-SERV-AUTHZ-001 (Basic)

2. Assessment objective
Verify that:

1) A default-deny policy is enforced for service accounts.
2) Service accounts can only perform actions explicitly authorized.

3. Assessment preparation

The assessment shall have access to:

- List of Hypervisor service accounts and service-facing interfaces.
- Documentation of the authorization model for service accounts (policies, defaults).
- At least one configuration with several service accounts, including one with no explicit permissions.

4. Assessment activities
The assessment shall at least:

1) Confirm from documentation/configuration that the default behaviour for service accounts is to deny actions not explicitly authorized.
2) Verify that a service account with no explicit permissions cannot perform management or data actions.
3) For a service account with a defined set of permissions, verify that allowed actions succeed while non-authorized actions are denied.

5. Assessment verdict

- Pass: default behaviour for service accounts is deny; an unprivileged service account cannot perform actions; and only explicitly authorized actions are permitted.
- Fail: unspecified actions for service accounts are allowed; or an unprivileged service account can perform operations.

6. Assessment evidence

- [SCOPE] List of service accounts and relevant interfaces.
- [CONFIG] Service-account authorization and default policy configuration.
- [DESIGN] Description of the service-account authorization model and default-deny behaviour.
- [LOG] / [OBSERVATION] Evidence of denied operations for unprivileged service accounts and permitted operations only when explicitly authorized.

6.3.1.11.2 Assessment Case AC-H-SERV-AUTHZ-002

1. Assessment reference
Requirement: REQ-H-SERV-AUTHZ-002 (Elevated)

2. Assessment objective
Verify that:

1) Granular authorization is enforced for service accounts, binding allowed actions to their authenticated identity.
2) Each service account is configured with least-privilege permissions.

3. Assessment preparation

The assessment shall have access to:

- Documentation on how service identities are represented and bound to permissions.
- Documentation on the permission/role model for service accounts.
- At least one configuration with multiple service accounts having different, scoped permission sets.

4. Assessment activities
The assessment shall at least:

1) Confirm that authorization decisions for service accounts are based on the authenticated service identity and not only on generic properties (e.g. IP or network zone).

2) For at least two service accounts with different permission sets, verify that each can only perform the actions explicitly bound to its identity and that actions assigned to another service identity are denied.

3) Verify that a service account configured with minimal permissions cannot perform operations outside that minimal set (least privilege).

5. Assessment verdict

- Pass: authorization for service accounts is identity-bound and granular; and least-privilege permission sets can be configured and are enforced.
- Fail: actions are not clearly bound to service identity; or service accounts can perform operations outside their defined permissions; or least-privilege cannot be demonstrated.

6. Assessment evidence

- [CONFIG] Service-account authorization configuration showing bindings between specific identities and allowed actions, including minimal-permission accounts.
- [DESIGN] Documentation of the service-account authorization model, identity binding and least-privilege approach.
- [LOG] / [OBSERVATION] Evidence of allowed operations within each service account's permissions and denied operations outside that scope.

6.3.1.11.3 Assessment Case AC-H-SERV-AUTHZ-003

1. Assessment reference
Requirement: REQ-H-SERV-AUTHZ-003 (Advanced)

2. Assessment objective
Verify that:

1) Fine-grained, policy-driven authorization is enforced for service accounts.

2) Permissions are bound to cryptographic service identities.

3) Context-aware enforcement (e.g. namespace, cluster, labels or similar attributes) is supported for service-account authorization decisions.

4) Integration with external authorization systems is supported where available.

3. Assessment preparation
The assessment shall have access to:

- Documentation of policy-driven authorization for service accounts (policy language or rules, conditions, effects).
- Documentation on how cryptographic service identities (e.g. certificates, tokens) are bound to permissions.
- Documentation on context attributes available for policy decisions.
- Documentation on integration with external authorization systems (e.g. external Policy Decision Point (PDP) or policy engine), if supported.
- At least one configuration where:
  - service-account policies reference cryptographic identities and at least one context attribute; and
  - where supported, an external authorization system is integrated.

4. Assessment activities
The assessment shall at least:

1) Verify that service-account authorization is expressed as policies (not just fixed roles) and that permissions for at least one service account are defined via such policies.

2) Verify that policies can reference cryptographic identifiers of service accounts and that, in the assessed configuration, authorization decisions for at least one service depend on its cryptographic identity.

3) Verify that at least one context attribute (e.g. namespace, cluster, workload label) can influence authorization decisions for service accounts and that access is allowed or denied based on that context.

4) Where integration with external authorization is supported, verify that the Hypervisor can consult or delegate service-account authorization decisions to an external system and that policy changes in that system are reflected in authorization outcomes.

5. Assessment verdict

- Pass: service-account authorization is fine-grained and policy-driven; permissions are bound to cryptographic service identities; context-aware enforcement is supported; and, where applicable, external authorization integration is effective.
- Fail: policies cannot be used to express fine-grained authorization; or permissions are not bound to cryptographic identities; or context cannot meaningfully influence authorization decisions; or supported external authorization integration does not work as intended.

6. Assessment evidence

- [CONFIG] Policy definitions and configuration for service-account authorization referencing cryptographic identities and context attributes, and, where applicable, external authorization integration settings.
- [DESIGN] Documentation of the policy-driven service authorization architecture, including cryptographic identity binding, context handling and external authorization interfaces.
- [LOG] / [OBSERVATION] Evidence of authorization decisions depending on cryptographic identity and context, and of changes resulting from external authorization policy updates where supported.

6.3.1.12 Assessment for Confidentiality Protection

6.3.1.12.1 Assessment Case AC-H-CONF-001

1. Assessment reference
Requirement: REQ-H-CONF-001 (Basic)

2. Assessment objective
Verify that:

1) the Hypervisor provides access-control mechanisms to restrict access, through Hypervisor functions and interfaces, to sensitive data associated with the Hypervisor product, including guest VM images, snapshot data, configuration files, credentials, and security-relevant logs; and

2) access to such sensitive data through Hypervisor functions and interfaces is permitted only to authenticated and authorized entities through controlled interfaces.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the categories of sensitive data managed or exposed through Hypervisor functions and interfaces and the controlled interfaces through which such data can be accessed or managed.
- Documentation describing the access-control model and the roles or identities authorized to access each relevant category of sensitive data through Hypervisor functions and interfaces.
- A test configuration with at least one authorized account and at least one unauthorized or less-privileged account, together with access to the relevant Hypervisor functions and interfaces.

4. Assessment activities
The assessment shall at least:

1) Review documentation and configuration to identify the categories of sensitive data in scope and the Hypervisor functions and interfaces through which the data is stored, managed, or exposed.

2) Using an authenticated and authorized entity, access representative items of the relevant sensitive-data categories through the documented controlled interfaces and verify that the access behaves as documented.

3) Using an unauthenticated, unauthorized, or less-privileged context, attempt to access the same categories of sensitive data through the Hypervisor functions and interfaces and verify that such access is denied.

5. Assessment verdict

- Pass: the Hypervisor restricts access, through its functions and interfaces, to the relevant categories of sensitive data; and only authenticated and authorized entities can access such data through the controlled interfaces.
- Fail: sensitive data can be accessed through Hypervisor functions or interfaces by unauthenticated or unauthorized entities; or the documented access-control model is not enforced through the relevant controlled interfaces.

6. Assessment evidence

- [DESIGN] Documentation of the sensitive-data categories in scope, the Hypervisor functions and interfaces that expose them, and the associated access-control model.
- [CONFIG] Hypervisor configuration showing permissions or policies for the relevant functions and interfaces.
- [LOG] / [OBSERVATION] Evidence of successful authorized access and denied unauthorized access through the Hypervisor functions and interfaces in scope.

6.3.1.12.2 Assessment Case AC-H-CONF-002

1. Assessment reference
Requirement: REQ-H-CONF-002 (Elevated)

2. Assessment objective
Verify that:

1) The Hypervisor supports encryption of sensitive data at rest, including VM images, snapshot data, configuration files, credentials, and security-relevant logs, as specified by the product.

2) When data-at-rest protection is enabled, encryption is effectively applied to the configured sensitive data repositories.

3) Encryption keys used for data-at-rest protection are stored and managed in a manner that prevents access by guest workloads and other unprivileged components.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing:
  - the data-at-rest encryption design, including which types of data and storage locations can be encrypted (VM images, snapshots, configuration files, credentials, security logs);
  - supported key-management approaches (for example internal key store, external KMS, hardware-backed keys);
  - how encryption and key management are configured and monitored by the administrator;
  - restrictions on access to encryption keys and key-management interfaces.
- A test configuration with:
  - one or more data stores configured to use the Hypervisor data-at-rest encryption mechanisms for the relevant categories of sensitive data;

- at least one administrative account authorized to configure and monitor encryption and key management;
- at least one guest workload and one non-privileged host context that can be used to attempt access to encrypted data and key material;
- where applicable, access to the underlying storage devices or files to inspect whether data is stored encrypted.

4. Assessment activities

The assessment shall at least:

1) Review documentation and configuration to identify:

- which sensitive data categories and storage locations are covered by data-at-rest encryption;
- how encryption is enabled and managed (for example per datastore, per volume, per file);
- where encryption keys are stored and how they are protected from guest and non-privileged access.

2) With the Hypervisor configured according to product guidance:

- create or update representative VM images, snapshots, configuration files, credentials, and security logs under the encrypted configuration;
- inspect the underlying storage (for example by viewing raw blocks, file contents from outside the Hypervisor context, or using vendor-provided diagnostic tools) to confirm that these items are stored in an encrypted form and not as plaintext;
- verify that disabling or removing access to encryption keys (for example by unmounting the key store or revoking KMS access) prevents normal decryption and access to the protected data.

3) From guest workloads and other non-privileged contexts:

- attempt to access encryption keys or key stores using available interfaces, filesystems, or APIs;
- verify that such attempts are denied and that key material is not readable or exportable by these contexts.

4) Where external or hardware-backed key management is used, verify that:

- keys used for data-at-rest encryption are stored in the external or hardware-backed mechanism as documented;
- the Hypervisor only receives derived or wrapped keys as documented and does not expose raw key material to guest or non-privileged components.

5. Assessment verdict

- Pass: the Hypervisor supports and correctly applies encryption for the configured categories of sensitive data at rest; encrypted data is not stored in plaintext on underlying storage; and encryption keys are managed and protected so that guest workloads and other unprivileged components cannot access them.
- Fail: sensitive data at rest configured to be protected is stored unencrypted or in recoverable plaintext; encryption keys are accessible to guest workloads or other unprivileged contexts; or the implemented behaviour differs from the documented encryption and key-management model.

6. Assessment evidence

- [DESIGN] Documentation of the data-at-rest encryption model, including covered data types, key-management design, and protection measures for key material.
- [CONFIG] Hypervisor configuration showing enabled encryption for relevant datastores and the configured key-management settings.
- [LOG] Events or audit trails related to enabling/disabling encryption, key-management operations, and failed access attempts to protected data or keys.

- [OBSERVATION] Test records and storage inspections (for example hex dumps, tool outputs) demonstrating that protected data is stored encrypted and that data becomes inaccessible when keys are revoked.
- [PLT-CAP] Evidence of any platform or external key-management capabilities used to support data-at-rest encryption.

6.3.1.12.3 Assessment Case AC-H-CONF-003

1. Assessment reference
Requirement: REQ-H-CONF-003 (Elevated)

2. Assessment objective
Verify that the Hypervisor supports encrypted transport for management interfaces, VM migration traffic, and Hypervisor-mediated inter-VM traffic where such communication paths are supported by the declared Hypervisor product, and that fallback or downgrade to unencrypted or weakly protected channels is prevented when encrypted transport is configured.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying Hypervisor management interfaces, VM migration paths, and Hypervisor-mediated inter-VM communication paths supported by the declared product.
- Documentation describing encrypted transport mechanisms supported for the relevant communication paths.
- A test configuration with:

  a) at least one management interface accessible over a network, where such an interface is part of the declared Hypervisor product;

  b) at least one VM migration path between Hypervisor hosts, where VM migration is supported by the declared Hypervisor product;

  c) at least two VMs whose inter-VM traffic can be mediated by the Hypervisor virtual networking layer, where such mediation is supported by the declared Hypervisor product;

  d) tools to capture and inspect network traffic on the relevant assessed communication paths.

4. Assessment activities
The assessment shall at least:

1) Identify the relevant communication paths supported by the declared Hypervisor product, including management interfaces, VM migration traffic, and Hypervisor-mediated inter-VM traffic, as applicable.

2) Configure encrypted transport for the relevant communication paths supported by the declared Hypervisor product, including as applicable:

- a representative management interface;
- a VM migration operation between Hypervisor hosts, where VM migration is supported;
- an inter-VM communication path mediated by the Hypervisor virtual networking layer, where such mediation is supported.

3) Use traffic capture, configuration inspection, protocol inspection, or equivalent evidence to verify that the relevant communication paths use encrypted transport when configured.

4) Attempt fallback or downgrade to unencrypted or weakly protected channels, where such fallback or downgrade attempt is feasible in the test environment and verify that fallback or downgrade is prevented when encrypted transport is configured.

5. Assessment verdict

- Pass: encrypted transport is supported and effective for the relevant management, migration, and Hypervisor-mediated inter-VM communication paths supported by the declared Hypervisor product, and

fallback or downgrade to unencrypted or weakly protected channels is prevented when encrypted transport is configured.

- Fail: encrypted transport is not supported for a relevant communication path supported by the declared Hypervisor product; or encrypted transport is configured but not effective; or fallback or downgrade to unencrypted or weakly protected channels is possible when encrypted transport is configured.

6. Assessment evidence

- [DESIGN] Documentation identifying the Hypervisor management, VM migration, and Hypervisor-mediated inter-VM communication paths supported by the declared product.
- [CONFIG] Configuration of encrypted transport for the relevant supported communication paths.
- [LOG] / [OBSERVATION] Traffic-capture, protocol-inspection, configuration-inspection, or equivalent evidence showing that configured encrypted communication is used on the relevant supported communication paths.
- [LOG] / [OBSERVATION] Evidence that fallback or downgrade to unencrypted or weakly protected channels is prevented when encrypted transport is configured.

6.3.1.12.4 Assessment Case AC-H-CONF-004

1. Assessment reference
Requirement: REQ-H-CONF-004 (Advanced)

2. Assessment objective
Verify that the Hypervisor supports provisioning and management of confidential-execution environments for guest workloads in which confidential guest workload memory and associated processor state are protected, during execution, from observation, inspection, or access through the Hypervisor, the host operating system, or administrative interfaces beyond what is documented in the confidential-execution model.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the Hypervisor confidential-execution architecture, including provisioning, management, and runtime protection of confidential guest workloads.
- Documentation identifying the security properties provided by the confidential-execution model, including restrictions on Hypervisor, host operating system, and administrative access to confidential guest memory and associated processor state.
- Documentation identifying any dependency on platform mechanisms, hardware-assisted isolation, firmware, Host OS services, or external trust services outside the declared product boundary or outside the responsibility of the manufacturer, together with the expected security properties.
- A test configuration with at least one guest VM that can be instantiated in confidential-execution mode on a platform and declared product configuration supporting the mechanisms required by REQ-H-CONF-004.
- Tools or interfaces available in the assessed environment to attempt observation, inspection, or access to confidential guest memory or associated processor state.

4. Assessment activities
The assessment shall at least:

1) Review the documented Hypervisor confidential-execution model and identify the security properties claimed for confidential guest workloads.
2) Review the documented behaviour of non-confidential workloads only where such comparison is relevant to the product's confidential-execution model. This comparison shall not be required where the product does not support an equivalent non-confidential mode or where such inspection capability is not provided by the declared product.
3) Instantiate a guest VM in confidential-execution mode and:

- verify that the workload is identified as running in confidential-execution mode according to the product documentation;
- attempt, using available Hypervisor, host, or administrative interfaces within the assessed environment, to observe, inspect, or access confidential guest memory or associated processor state;
- verify that such observation, inspection, or access is prevented or results only in protected or unintelligible data, in accordance with the documented confidential-execution model;
- verify that any dependency on platform mechanisms, hardware-assisted isolation, firmware, Host OS services, or external trust services outside the declared product boundary is identified as an Operational Environment dependency together with the expected security properties.

5. Assessment verdict

- Pass: the Hypervisor supports provisioning and management of confidential-execution environments for guest workloads; when such environments are used, confidential workload memory and associated processor state cannot be observed, inspected, or accessed through Hypervisor, host operating system, or administrative interfaces beyond what is documented in the confidential-execution model; and dependencies on platform mechanisms, hardware-assisted isolation, firmware, Host OS services, or external trust services outside the declared product boundary are identified with their expected security properties.
- Fail: the Hypervisor does not support provisioning or management of confidential-execution environments as required by REQ-H-CONF-004; or workloads configured as confidential can have their confidential memory or associated processor state observed, inspected, or accessed in clear form through Hypervisor, host operating system, or administrative interfaces contrary to the documented confidential-execution model; or dependencies outside the declared product boundary are not identified.

6. Assessment evidence

- [DESIGN] Documentation of the confidential-execution architecture, including the supported confidential-execution model, provisioning and management workflow, and restrictions on Hypervisor, host operating system, and administrative access.
- [CONFIG] Hypervisor and platform configuration showing how confidential execution is enabled and which guest workloads are configured to use it.
- [OE] Documentation identifying dependencies on platform mechanisms, hardware-assisted isolation, firmware, Host OS services, or external trust services outside the declared product boundary, including expected security properties.
- [LOG] / [OBSERVATION] Evidence of creation, configuration, and management of a confidential guest workload.
- [OBSERVATION] Evidence that attempts to observe, inspect, or access confidential guest memory or associated processor state through Hypervisor, host operating system, or administrative interfaces are prevented or result only in protected or unintelligible data, in accordance with the documented confidential-execution model.
- [PLT-CAP] Evidence of platform confidential-execution capabilities used by the Hypervisor, where applicable.

6.3.1.13 Assessment for Availability and Resilience

6.3.1.13.1 Assessment Case AC-H-AVAIL-001

1. Assessment reference
Requirement: REQ-H-AVAIL-001 (Basic)

2. Assessment objective
Verify that:

1) the Hypervisor detects failures of internal management and control-plane components;

2) the Hypervisor supports recovery of those components at the component level, without initiating a full host reboot, except where the nature of the failure precludes component-level recovery without loss of system integrity; and

3) the Hypervisor maintains the operational state of running guest virtual machines during and after the component-level recovery process.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the internal management and control-plane components, their failure-detection mechanisms, and the supported component-level recovery methods.
- Documentation identifying any failure modes for which the product design permits only full host reboot as the safe recovery path.
- A test configuration with running guest virtual machines and the ability to induce or simulate failure of at least one internal management or control-plane component.

4. Assessment activities
The assessment shall at least:

1) Review documentation to identify internal management and control-plane components, the mechanisms used to detect their failure, the supported component-level recovery methods, and any documented exceptions requiring full host reboot to preserve system integrity.

2) In the assessed configuration, induce or simulate failure of at least one internal management or control-plane component for which component-level recovery is supported and verify that the Hypervisor detects the failure.

3) Verify that the Hypervisor recovers the affected component at the component level, without initiating a full host reboot, and that the operational state of running guest virtual machines is maintained during and after the recovery process.

4) Where the product documentation identifies a failure mode for which full host reboot is the only safe recovery path, verify that this exception is explicitly documented as part of the product design.

5. Assessment verdict

- Pass: the Hypervisor detects failures of internal management and control-plane components; supports component-level recovery without full host reboot except for documented failure modes where such recovery would compromise system integrity; and maintains the operational state of running guest virtual machines during and after the component-level recovery process.
- Fail: failures of internal management or control-plane components are not detected; recovery requires a full host reboot for cases where component-level recovery is claimed to be supported; documented exceptions are absent or inconsistent with observed behaviour; or the operational state of running guest virtual machines is not maintained during and after the recovery process.

6. Assessment evidence

- [CONFIG] Configuration of management/control-plane components and their failure-detection and recovery mechanisms.
- [DESIGN] Documentation of the management/control-plane architecture, component-level recovery behaviour, and documented reboot-only exceptions.
- [LOG] / [OBSERVATION] Evidence of failure detection, component-level recovery without host reboot where applicable, and maintenance of running guest virtual machines during and after recovery.

6.3.1.13.2 Assessment Case AC-H-AVAIL-002

1. Assessment reference
Requirement: REQ-H-AVAIL-002 (Elevated)

2. Assessment objective
Verify that:

1) The Hypervisor provides mechanisms to support high availability of guest virtual machines.

2) Host or Hypervisor service failure is automatically detected.

3) Affected guest VMs are automatically restarted or migrated to another available host using native clustering or integration with external clustering/orchestration systems.

3. Assessment preparation
The assessment shall have access to:

- Documentation of Hypervisor high-availability mechanisms for guest VMs, including clustering and integrations with external orchestrators.
- Documentation describing detection of host/Hypervisor service failure and the associated response.
- A test environment with at least two Hypervisor hosts participating in a cluster or managed by an orchestrator, hosting one or more test VMs.

4. Assessment activities
The assessment shall at least:

1) Confirm that high-availability or clustering features are enabled/configured for selected guest VMs.

2) Induce or simulate failure of a Hypervisor host or critical Hypervisor service (e.g. stopping the Hypervisor service or isolating the host) and verify that the failure is automatically detected by the HA/cluster/orchestration mechanism.

3) Verify that affected guest VMs are automatically restarted or migrated to another available host, without requiring manual intervention, and that the VMs become available again after the recovery action.

5. Assessment verdict

- Pass: host/Hypervisor service failures are automatically detected; guest VMs configured for HA are automatically restarted or migrated to another host using native or integrated clustering/orchestration; and availability of such VMs is restored automatically.
- Fail: failures are not automatically detected; or affected guest VMs are not automatically restarted/migrated; or manual intervention is required to restore availability in contradiction with the documented HA behaviour.

6. Assessment evidence

- [CONFIG] High-availability/clustering configuration for guest VMs and host group/cluster settings.
- [DESIGN] Documentation of HA/cluster/orchestration architecture and failure-handling behaviour.
- [LOG] / [OBSERVATION] Evidence of failure detection and subsequent automatic restart or migration of guest VMs.

6.3.1.13.3 Assessment Case AC-H-AVAIL-003

1. Assessment reference
Requirement: REQ-H-AVAIL-003 (Advanced)

2. Assessment objective
Verify that the Hypervisor supports continuity-preserving mechanisms for selected guest VMs, such as live migration, restart from a trusted state, state replication, clustering integration, or equivalent continuity mechanisms, and that where these mechanisms depend on Host OS, storage, clustering, platform, or external lifecycle-management mechanisms outside the declared product boundary, those mechanisms are documented as Operational Environment dependencies.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the Hypervisor continuity-preserving mechanisms for guest VMs, including supported mechanisms such as live migration, restart from a trusted state, state replication, clustering integration, or equivalent mechanisms.
- Documentation identifying the guest VM categories or configurations for which continuity-preserving mechanisms are supported.
- Documentation describing the expected interruption, recovery behaviour, limitations, and assumptions of the supported continuity-preserving mechanisms.
- Documentation identifying any Host OS, storage, clustering, platform, or external lifecycle-management mechanisms outside the declared product boundary or outside the responsibility of the manufacturer that are required for the continuity-preserving mechanism, together with the expected security properties.
- A test or demonstration environment supporting at least one continuity-preserving mechanism declared by the Hypervisor.

4. Assessment activities

The assessment shall at least:

1) Review the documented Hypervisor continuity-preserving mechanisms and identify the supported mechanisms relevant to REQ-H-AVAIL-003.
2) Configure a selected guest VM to use at least one documented continuity-preserving mechanism.
3) Simulate or initiate a relevant failure, relocation, restart, or recovery condition according to the documented mechanism.
4) Verify that the guest VM is recovered, relocated, restarted, or continued according to the documented continuity behaviour.
5) Verify that the observed interruption and recovery behaviour are consistent with the documented mechanism and assumptions.
6) Where the mechanism depends on Host OS, storage, clustering, platform, or external lifecycle-management mechanisms outside the declared product boundary or outside the responsibility of the manufacturer, verify that these dependencies are identified as Operational Environment dependencies together with their expected security properties.

5. Assessment verdict

- Pass: the Hypervisor supports at least one documented continuity-preserving mechanism for selected guest VMs; the mechanism operates as documented in the assessed configuration; observed recovery, relocation, restart, or continuity behaviour is consistent with the documented assumptions; and any required Host OS, storage, clustering, platform, or external lifecycle-management dependencies outside the declared product boundary are documented as Operational Environment dependencies with expected security properties.
- Fail: the Hypervisor does not support the continuity-preserving mechanism required by REQ-H-AVAIL-003; or the supported mechanism does not operate as documented; or observed interruption or recovery behaviour is inconsistent with the documented assumptions; or required Operational Environment dependencies are not identified where the mechanism depends on components outside the declared product boundary or outside the responsibility of the manufacturer.

6. Assessment evidence

- [DESIGN] Documentation of Hypervisor continuity-preserving mechanisms, including supported guest VM categories, limitations, and assumptions.
- [CONFIG] Configuration of the selected guest VM and continuity-preserving mechanism.
- [LOG] / [OBSERVATION] Evidence of recovery, relocation, restart, state replication, live migration, clustering integration, or equivalent continuity operation, as applicable to the declared mechanism.
- [OE] Documentation identifying Operational Environment dependencies and expected security properties for continuity-preserving mechanisms, where applicable.

6.3.1.14 Assessment for Logging

6.3.1.14.1 Assessment Case AC-H-LOG-001

1. Assessment reference
Requirement: REQ-H-LOG-001 (Basic)

2. Assessment objective
Verify that the Hypervisor:

1) generates time-stamped audit logs for security-relevant administrative actions and security events, including at least authentication attempts, configuration changes, and virtual machine lifecycle operations; and

2) includes in each audit record, as applicable to the event type, an event type identifier, the time of the event, the outcome of the event, and the identity of the associated user, service, or process.

3. Assessment preparation

The assessment shall have access to:

- Documentation of the Hypervisor logging and auditing capabilities and log formats.
- Documentation describing which actions and events are treated as security-relevant.
- A configuration with administrative access enabled and at least one VM that can be created, modified, started, stopped, and deleted.

4. Assessment activities
The assessment shall at least:

1) Review documentation and configuration to identify the audit log sources and where audit logs are stored.

2) Perform, in the assessed configuration, successful and failed authentication attempts, security-relevant configuration changes, and virtual machine lifecycle operations.

3) Verify that audit logs are generated for these actions and events and that each corresponding audit record includes, as applicable to the event type, an event type identifier, the time of the event, the outcome of the event, and the identity of the associated user, service, or process.

5. Assessment verdict

- Pass: the Hypervisor generates time-stamped audit logs for the required categories of security-relevant administrative actions and security events, and each corresponding audit record includes the required information as applicable to the event type.
- Fail: one or more required categories of actions or events are not logged; or audit records do not include the required information as applicable to the event type.

6. Assessment evidence

- [CONFIG] Logging and audit configuration for the Hypervisor.
- [DESIGN] Documentation of audit logging behaviour, log format, and logged event categories.
- [LOG] / [OBSERVATION] Sample audit log entries showing the required categories of actions and events and the required record elements.

6.3.1.14.2 Assessment Case AC-H-LOG-002

1. Assessment reference
Requirement: REQ-H-LOG-002 (Elevated)

2. Assessment objective
Verify that:

1) Audit logs are protected from unauthorized access, modification or deletion.

2) Access to audit logs is restricted to authorized administrative roles or processes.

3) The Hypervisor supports secure export of audit logs to external log management/SIEM systems using authenticated and encrypted channels.

3. Assessment preparation

The assessment shall have access to:

- Documentation on storage locations for audit logs and associated access controls (file permissions, roles, APIs).
- Documentation on roles/processes allowed to access or manage audit logs.
- Documentation of log export mechanisms and supported secure transport options (e.g. TLS-protected syslog, HTTPS, secure agents).
- A configuration with audit logging enabled and at least one external log management or SIEM endpoint (test instance) reachable via a secure channel.

4. Assessment activities
The assessment shall at least:

1) Review configuration and underlying platform permissions to confirm that audit log storage is restricted to authorized administrative roles/processes and that modification/deletion by non-authorized users or workloads is prevented.

2) Attempt to read, modify or delete audit logs using a non-privileged account or process and verify that these operations are denied.

3) Configure secure export of audit logs to an external log management/SIEM endpoint using an authenticated, encrypted channel as documented (e.g. TLS with server authentication and, where applicable, client authentication).

4) Verify, via configuration and protocol inspection, that the export occurs over authenticated and encrypted channels and that logs received by the external system correspond to those generated locally.

5. Assessment verdict

- Pass: audit logs are protected against unauthorized access/modification/deletion; only authorized administrative roles/processes can access them; and secure, authenticated and encrypted export to external log/SIEM systems is supported and functional.
- Fail: non-authorized access or tampering with audit logs is possible; or access controls on audit logs are insufficient; or log export uses unauthenticated/unencrypted channels or cannot be securely configured.

6. Assessment evidence

- [CONFIG] Audit log storage permissions, role mappings and export configuration (including security settings).
- [DESIGN] Documentation of audit log protection model and secure export mechanisms.
- [LOG] / [OBSERVATION] Evidence of denied access attempts by non-privileged accounts and successful secure export (e.g. protocol traces, external log records).

6.3.1.14.3 Assessment Case AC-H-LOG-003

1. Assessment reference
Requirement: REQ-H-LOG-003 (Advanced)

2. Assessment objective
Verify that the Hypervisor supports mechanisms to protect audit logs generated under REQ-H-LOG-001 against undetected modification, deletion, or reordering, and that the mechanism enables verification of the integrity and time or sequence of security-relevant audit records using cryptographic protection, trusted time-stamping, append-only or tamper-evident storage, secure export to an external log management system, or an equivalent mechanism.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the mechanism used to protect Hypervisor audit logs against undetected modification, deletion, or reordering.
- Documentation identifying whether the mechanism uses cryptographic protection, trusted time-stamping, append-only or tamper-evident storage, secure export to an external log management system, or an equivalent mechanism.
- Documentation describing how integrity and time or sequence of security-relevant audit records can be verified.
- Where secure export to an external log management system or another mechanism outside the declared product boundary is used, documentation identifying the corresponding Operational Environment dependency and expected security properties.
- A test or demonstration configuration with Hypervisor audit logging enabled and the log-protection mechanism configured.
- Tools or interfaces to generate audit records and to attempt modification, deletion, reordering, or equivalent tampering of audit records.

4. Assessment activities
The assessment shall at least:

1) Review the documented log-protection mechanism and identify how it protects against undetected modification, deletion, or reordering.
2) Generate representative Hypervisor audit records covered by REQ-H-LOG-001.
3) Verify that the configured mechanism enables verification of the integrity and time or sequence of the generated audit records.
4) Attempt to modify, delete, reorder, or otherwise tamper with audit records, or simulate such tampering, and verify that the tampering is prevented or detectable.
5) Where secure export to an external log management system or another mechanism outside the declared product boundary is used, verify that the Operational Environment dependency and its expected security properties are documented.

5. Assessment verdict

- Pass: the Hypervisor supports a mechanism that protects audit logs generated under REQ-H-LOG-001 against undetected modification, deletion, or reordering; the mechanism enables verification of the integrity and time or sequence of security-relevant audit records; tampering is prevented or detectable; and any external log-management or Operational Environment dependency is documented with expected security properties.
- Fail: the Hypervisor does not support a mechanism to protect audit logs against undetected modification, deletion, or reordering; or integrity and time or sequence of security-relevant audit records cannot be verified; or tampering is not prevented and is not detectable; or an external dependency required for log protection is not documented.

6. Assessment evidence

- [DESIGN] Documentation of the Hypervisor audit-log protection mechanism, including integrity and time or sequence verification.

- [CONFIG] Configuration of cryptographic protection, trusted time-stamping, append-only or tamper-evident storage, secure external log export, or equivalent log-protection mechanism.
- [LOG] / [OBSERVATION] Evidence that generated audit records are protected and that integrity and time or sequence can be verified.
- [LOG] / [OBSERVATION] Evidence that modification, deletion, reordering, or equivalent tampering is prevented or detectable.
- [OE] Documentation identifying external log-management or other Operational Environment dependencies and expected security properties, where applicable.

6.3.1.15 Assessment for Secure Update

6.3.1.15.1 Assessment Case AC-H-UPD-001

1. Assessment reference
Requirement: REQ-H-UPD-001 (Basic)

2. Assessment objective
Verify that:

1) The Hypervisor is delivered in a form that supports authenticity and integrity verification of Hypervisor updates before installation or activation.
2) The Hypervisor supports authenticity and integrity verification of Hypervisor updates using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
3) Where the Hypervisor includes the mechanism that verifies, installs, or activates its updates, this mechanism prevents installation or activation of an update when authenticity or integrity verification fails and records a security-relevant event.
4) Where update delivery, verification, installation, or activation is performed by a Host OS update service, package manager, platform update service, appliance update mechanism, M&O system, orchestration-based update pipeline, or other Operational Environment mechanism outside the declared Hypervisor product or outside the responsibility of the manufacturer, the product security documentation identifies that mechanism, the expected security properties, and the assumptions required for secure update of the Hypervisor.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying the Hypervisor components delivered as part of the declared Hypervisor product.
- Documentation describing the update artefacts applicable to those Hypervisor components.
- Documentation describing how authenticity and integrity verification of Hypervisor updates is supported before installation or activation, including the verification metadata used, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Documentation identifying whether update delivery, verification, installation, or activation is performed by the Hypervisor, by another component under the responsibility of the manufacturer, or by an Operational Environment mechanism.
- Where the Hypervisor includes the mechanism that verifies, installs, or activates updates, a test configuration allowing use of a valid update artefact and an update artefact with invalid, missing, revoked, or intentionally modified authenticity or integrity protection, or an equivalent test condition.
- Where update delivery, verification, installation, or activation is performed by an Operational Environment mechanism, product security documentation identifying the expected update mechanism, expected security properties, and assumptions required for secure update of the Hypervisor.

4. Assessment activities
The assessment shall at least:

1) Identify the Hypervisor components and update artefacts covered by REQ-H-UPD-001.

2) Check that the Hypervisor is delivered in a form that supports authenticity and integrity verification of Hypervisor updates before installation or activation.

3) Check that the update artefacts support authenticity and integrity verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.

4) Check whether update delivery, verification, installation, or activation is performed by the Hypervisor, by another component under the responsibility of the manufacturer, or by an Operational Environment mechanism.

5) Where the Hypervisor includes the mechanism that verifies, installs, or activates updates, attempt to install or activate an update whose authenticity or integrity verification fails and verify that installation or activation is prevented and that a security-relevant event is recorded.

6) Where update delivery, verification, installation, or activation is performed outside the declared Hypervisor product or outside the responsibility of the manufacturer, verify that the product security documentation identifies the Operational Environment dependency, the expected security properties, and the assumptions required for secure update of the Hypervisor.

5. Assessment verdict

- Pass: the Hypervisor is delivered in a form that supports authenticity and integrity verification of Hypervisor updates before installation or activation; update artefacts support verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata; and, where the Hypervisor includes the mechanism that verifies, installs, or activates updates, failed authenticity or integrity verification prevents installation or activation and is recorded as a security-relevant event. Where update delivery, verification, installation, or activation is performed outside the declared Hypervisor product or outside the responsibility of the manufacturer, the Operational Environment dependency, expected security properties, and assumptions are documented.

- Fail: the Hypervisor is not delivered in a form that supports authenticity and integrity verification of Hypervisor updates before installation or activation; or the update artefacts do not support the required verification metadata; or, where the Hypervisor includes the mechanism that verifies, installs, or activates updates, an update whose authenticity or integrity verification fails can be installed or activated, or the failure is not recorded as a security-relevant event; or the relevant Operational Environment dependency is not documented where update delivery, verification, installation, or activation is performed outside the declared Hypervisor product or outside the responsibility of the manufacturer.

6. Assessment evidence

- [DESIGN] Documentation of the Hypervisor update model, including update artefacts and the component or mechanism responsible for update delivery, verification, installation, or activation.

- [DESIGN] Documentation of authenticity and integrity verification metadata used for Hypervisor updates, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.

- [CONFIG] Configuration of the Hypervisor update mechanism, where the Hypervisor includes the mechanism that verifies, installs, or activates updates.

- [LOG] / [OBSERVATION] Where the Hypervisor includes the mechanism that verifies, installs, or activates updates, evidence that a valid update is accepted and that an update with failed authenticity or integrity verification is prevented from installation or activation and recorded as a security-relevant event.

- [GUIDANCE] / [OE] Where update delivery, verification, installation, or activation is performed outside the declared Hypervisor product or outside the responsibility of the manufacturer, product security documentation identifying the Operational Environment dependency, expected security properties, and assumptions required for secure update of the Hypervisor.

6.3.1.15.2 Assessment Case AC-H-UPD-002

1. Assessment reference
Requirement: REQ-H-UPD-002 (Elevated)

2. Assessment objective
Verify that:

1) The Hypervisor supports rollback protection for Hypervisor updates where rollback protection is provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer.

2) Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed Hypervisor updates is prevented.

3) Authorized rollback to a previously trusted Hypervisor version is only permitted when:

a) the rollback image or package is verified for integrity and authenticity;

b) the rollback action is explicitly authorized by an administrator with appropriate privileges or an equivalent authorized control function; and

c) the rollback operation is recorded as a security-relevant event.

3. Assessment preparation

The assessment shall have access to:

- Documentation identifying whether rollback protection for Hypervisor updates is provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer.
- Documentation describing the rollback protection mechanism, where rollback protection is provided.
- Documentation describing procedures for authorized rollback to a previously trusted Hypervisor version, where authorized rollback is supported.
- Documentation identifying the administrator privileges or equivalent authorized control function required to authorize rollback, where authorized rollback is supported.
- Documentation describing how rollback operations are recorded as security-relevant events, where rollback is supported.
- A test or demonstration configuration allowing an outdated, revoked, or replayed Hypervisor update to be attempted, where rollback protection is provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer.
- A test or demonstration configuration allowing authorized rollback to a previously trusted Hypervisor version, where authorized rollback is supported.

4. Assessment activities
The assessment shall at least:

1) Check whether rollback protection for Hypervisor updates is provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer.

2) Where rollback protection is not provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer, verify that the product security documentation identifies the update and rollback assumptions applicable to the declared product configuration.

3) Where rollback protection is provided, attempt to install an outdated, revoked, or replayed Hypervisor update outside the documented authorized rollback procedure and verify that installation is prevented.

4) Where authorized rollback is supported, perform an authorized rollback using the documented procedure and verify that:

a) the rollback image or package is verified for integrity and authenticity;

b) the rollback action requires explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; and

c) the rollback operation is recorded as a security-relevant event.

5. Assessment verdict

- Pass: where rollback protection is provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer, unauthorized installation of outdated, revoked, or replayed Hypervisor updates is prevented; and authorized rollback to a previously trusted Hypervisor version is permitted only following integrity and authenticity verification of the rollback image or package, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function, and recording of the rollback operation as a security-relevant event. Where rollback protection is not provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer, the update and rollback assumptions applicable to the declared product configuration are documented.
- Fail: where rollback protection is provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer, outdated, revoked, or replayed Hypervisor updates can be installed without authorization; or authorized rollback can be performed without integrity and authenticity verification; or rollback can be performed without explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; or the rollback operation is not recorded as a security-relevant event.

6. Assessment evidence

- [DESIGN] Documentation identifying whether rollback protection is provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer.
- [DESIGN] Documentation of rollback protection mechanisms and authorized rollback procedures, where rollback protection or authorized rollback is supported.
- [CONFIG] Configuration or policy information showing rollback authorization controls, where authorized rollback is supported.
- [LOG] / [OBSERVATION] Where rollback protection is provided, evidence that outdated, revoked, or replayed updates are prevented outside the authorized rollback procedure.
- [LOG] / [OBSERVATION] Where authorized rollback is supported, evidence that rollback is performed only after integrity and authenticity verification, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function, and recording as a security-relevant event.
- [GUIDANCE] / [OE] Where rollback protection is not provided by the Hypervisor or by an update mechanism under the responsibility of the manufacturer, documentation identifying update and rollback assumptions applicable to the declared product configuration.

6.3.1.15.3 Assessment Case AC-H-UPD-003

1. Assessment reference
Requirement: REQ-H-UPD-003 (Advanced)

2. Assessment objective
Verify that:

1) the declared product configuration identifies whether live patching of Hypervisor components is supported by the declared Hypervisor product;

2) where live patching of Hypervisor components is supported by the declared Hypervisor product, the Hypervisor provides configuration or administrative controls allowing the operator to enable live patching according to the operational security policy;

3) where live patching of Hypervisor components is supported by the declared Hypervisor product, the Hypervisor provides configuration or administrative controls allowing the operator to disable live patching according to the operational security policy;

4) where live patching of Hypervisor components is supported by the declared Hypervisor product, the Hypervisor provides configuration or administrative controls allowing the operator to restrict live patching according to the operational security policy;

5) where live patching of Hypervisor components is supported by the declared Hypervisor product, the product security documentation identifies security and operational trade-offs associated with live patching, including that live patching is not necessarily preferable to reboot-based update.

This assessment does not require execution of a live patch, demonstration of vulnerability remediation by live patching, or demonstration that live patching is preferable to reboot-based update.

3. Assessment preparation
The assessment shall have access to:

- Product security documentation identifying whether live patching of Hypervisor components is supported by the declared Hypervisor product;
- Product security documentation describing the configuration or administrative controls for live patching, where live patching is supported;
- Configuration or administrative interface documentation showing how live patching can be enabled, disabled, or restricted, where live patching is supported;
- Documentation describing how these controls support application of the operational security policy, where live patching is supported;
- Product security documentation describing security and operational trade-offs associated with live patching, including conditions where reboot-based update may be recommended or required, where live patching is supported.

4. Assessment activities
The assessment shall at least:

1) Check whether live patching of Hypervisor components is supported by the declared Hypervisor product.

2) If live patching of Hypervisor components is not supported by the declared Hypervisor product, check that this is identified in the product security documentation and proceed to the verdict based on the condition in REQ-H-UPD-003 not being triggered.

3) Where live patching is supported, check that the Hypervisor provides configuration or administrative controls to enable live patching.

4) Where live patching is supported, check that the Hypervisor provides configuration or administrative controls to disable live patching.

5) Where live patching is supported, check that the Hypervisor provides configuration or administrative controls to restrict live patching.

6) Where live patching is supported, check that the controls can be applied according to the operational security policy described in the product security documentation.

7) Where live patching is supported, check that the product security documentation identifies security and operational trade-offs associated with live patching.

8) Where live patching is supported, check that the product security documentation does not imply that live patching is always preferable to reboot-based update.

5. Assessment verdict

- Pass: live patching of Hypervisor components is not supported by the declared Hypervisor product and this is identified in the product security documentation.
- Pass: live patching of Hypervisor components is supported by the declared Hypervisor product and the required evidence demonstrates that the Hypervisor provides configuration or administrative controls allowing the operator to enable, disable, or restrict live patching according to the operational security policy, and that the product security documentation identifies the relevant security and operational trade-offs.

- Fail: the product security documentation does not identify whether live patching of Hypervisor components is supported by the declared Hypervisor product.
- Fail: live patching of Hypervisor components is supported by the declared Hypervisor product and the required evidence is missing or does not demonstrate that the Hypervisor provides configuration or administrative controls allowing the operator to enable, disable, or restrict live patching according to the operational security policy, or the product security documentation does not identify the relevant security and operational trade-offs.

6. Assessment evidence

- [DESIGN]:
  - Product security documentation identifying whether live patching of Hypervisor components is supported by the declared Hypervisor product.
  - Documentation explaining how live patching controls support the operational security policy, where live patching is supported.
  - Product security documentation describing security and operational trade-offs associated with live patching, where live patching is supported.
- [CONFIG] Configuration or administrative control documentation for enabling, disabling, and restricting live patching, where live patching is supported.
- [OBSERVATION] Evidence that the relevant controls are available through the documented configuration or administrative interfaces, where live patching is supported.

6.3.1.16 Assessment for Secure Configuration and Default

6.3.1.16.1 Assessment Case AC-H-CFG-001

1. Assessment reference
Requirement: REQ-H-CFG-001 (Basic)

2. Assessment objective
Verify that:

1) By default, the Hypervisor disables remote administrative access pathways and insecure management interfaces that are not required for initial provisioning (e.g. unauthenticated/weak remote shell, legacy or unencrypted management APIs, non-essential vendor diagnostic ports).
2) The Hypervisor provides configuration mechanisms that allow administrators to explicitly enable only those interfaces and services needed for the intended Virtualisation, orchestration and core host management operations.

3. Assessment preparation
The assessment shall have access to:

- Documentation listing Hypervisor administrative/management interfaces (remote shell, APIs, diagnostic ports, web consoles, etc.) and which are required for initial provisioning.
- Documentation describing configuration options to enable/disable interfaces and services.
- A fresh or default installation of the Hypervisor in its documented default configuration.

4. Assessment activities
The assessment shall at least:

1) In the default configuration, attempt to access all documented administrative and management interfaces (including remote shell, legacy APIs, diagnostic ports) and verify that only those required for initial provisioning are enabled; insecure or unnecessary interfaces are disabled by default.

2) Review configuration mechanisms (e.g. CLI, API, UI) and verify that administrators can explicitly enable specific interfaces/services; then enable a subset of interfaces required for typical Virtualisation/orchestration/host management operations and confirm:

- chosen interfaces become accessible; and

- interfaces/services not explicitly enabled remain disabled.

5. Assessment verdict

- Pass: non-essential and insecure administrative/management interfaces are disabled by default; and administrators can enable only those interfaces/services required for intended operations via explicit configuration.

- Fail: insecure or non-required administrative interfaces are enabled by default; or there is no practical way to selectively enable only necessary interfaces/services.

6. Assessment evidence

- [CONFIG] Default and modified configuration showing interface/service enablement states.

- [DESIGN] Documentation of management interfaces, default exposure, and configuration mechanisms.

- [OBSERVATION] / [LOG] Connection attempts and screenshots/CLI output showing which interfaces are reachable in default and explicitly enabled states.

6.3.1.16.2 Assessment Case AC-H-CFG-002

1. Assessment reference
Requirement: REQ-H-CFG-002 (Elevated)

2. Assessment objective
Verify that:

1) The Hypervisor validates configuration parameters before applying them.

2) Configuration validation logic detects and prevents configurations that would:

- disable or bypass security requirements defined in the present document; or

- violate guarantees for VM isolation, control-plane protection, or minimum resource allocations for security/management functions.

3) When a configuration change is rejected for security reasons, the Hypervisor provides clear feedback indicating the reason for rejection.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing configuration validation, including examples of invalid or disallowed settings (e.g. disabling mandatory controls, removing isolation, reducing reserved resources below minimums).

- Documentation listing mandatory security controls and key guarantees (VM isolation, control-plane protection, minimum security/management resources).

- A test environment where security-relevant configuration parameters can be changed.

4. Assessment activities
The assessment shall at least:

1) Identify a set of configuration parameters whose insecure values would:

- disable/bypass mandatory security controls;

- break VM isolation;

- undermine control-plane protection; or
- reduce reserved resources for security/management below documented minima.

2) Attempt to apply such insecure configurations and verify that the Hypervisor rejects or blocks them based on configuration validation.

3) Observe the feedback presented to the administrator for each rejected change and verify that it clearly indicates the security-related reason (e.g. "would disable mandatory control X", "would violate minimum resource reservation for control plane", "would compromise VM isolation").

4) Apply valid configuration changes and verify that they are accepted and correctly enforced, demonstrating that validation is specific to insecure cases, not a generic failure.

5. Assessment verdict

- Pass: configuration parameters are validated before application; attempts to apply configurations that disable/bypass mandatory security controls or violate isolation/control-plane/minimum-resource guarantees are prevented; and clear feedback is provided for security-related rejections.
- Fail: unsafe configurations can be applied; or configuration validation does not cover the specified guarantees; or security-related rejection feedback is missing or unclear.

6. Assessment evidence

- [CONFIG] Examples of attempted insecure configurations and resulting states (accepted/rejected).
- [DESIGN] Documentation of configuration validation logic and mappings to mandatory controls and key guarantees.
- [LOG] / [OBSERVATION] CLI/UI output and error messages showing rejection of insecure configurations and the reasons given; evidence of accepted secure configuration changes.

6.3.1.16.3 Assessment Case AC-H-CFG-003

1. Assessment reference
Requirement: REQ-H-CFG-003 (Advanced)

2. Assessment objective
Verify that:

1) The Hypervisor supports defining and using security configuration baselines specifying secure settings for critical components, including at least:

- I/O Memory Management Unit (IOMMU) configuration;
- virtual switch policies;
- host firewall rules, as implemented within the Hypervisor product.

2) The Hypervisor can automatically validate current configuration against a selected security baseline.

3) The Hypervisor generates alerts or audit events when non-compliant settings are detected.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing security configuration baselines (how they are defined, stored, selected and applied).
- Documentation showing how baseline rules cover IOMMU configuration, virtual switch policies and host firewall rules within the Hypervisor product.
- Documentation of automatic baseline validation mechanisms and alert/audit behaviour on non-compliance.

- A test configuration where at least one security baseline can be defined/selected and where deviations from that baseline can be introduced.

4. Assessment activities
The assessment shall at least:

1) Define or select a security configuration baseline that includes explicit requirements for IOMMU settings, virtual switch policies and host firewall rules, according to product capabilities.

2) Ensure the Hypervisor configuration is aligned with the baseline and run the baseline validation function; verify that the Hypervisor reports compliance and does not generate non-compliance alerts/events.

3) Introduce one or more deliberate deviations from the baseline (e.g. relax a firewall rule, disable or weaken a virtual switch policy, misconfigure IOMMU-related settings) and run baseline validation again.

4) Verify that:

- the Hypervisor detects the non-compliant settings; and

- alerts or audit events are generated indicating which configuration elements are non-compliant with the selected baseline.

5. Assessment verdict

- Pass: security baselines can be defined/used for critical components (IOMMU, virtual switch, host firewall); baseline validation can be run automatically; and non-compliant settings trigger alerts or audit events identifying the deviations.

- Fail: security baselines cannot be defined or do not cover the required components; automatic validation against a baseline is not available; or non-compliance does not result in clear alerts/audit events.

6. Assessment evidence

- [CONFIG] Example security baseline definition and corresponding Hypervisor configuration (before and after deviation).

- [DESIGN] Documentation of security baseline management, validation logic and alert/audit integration.

- [LOG] / [OBSERVATION] Outputs from baseline validation showing compliant and non-compliant states, including generated alerts or audit log entries identifying non-compliant settings.

6.3.1.17 Assessment for Data Minimisation

6.3.1.17.1 Assessment Case AC-H-DM-001

1. Assessment reference
Requirement: REQ-H-DM-001 (Basic)

2. Assessment objective
Verify that Hypervisor logs, telemetry, and diagnostics do not include plaintext authentication credentials, full cryptographic keys, guest workload payload data, or VM memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data, unless explicitly required for a documented diagnostic function and protected according to the product security documentation.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing Hypervisor logging, telemetry, and diagnostic functions.

- Documentation identifying categories of data collected by Hypervisor logs, telemetry, and diagnostics.

- Documentation identifying any diagnostic functions that intentionally collect sensitive data and the protections applied to such data.

- A test or demonstration configuration enabling representative Hypervisor logs, telemetry, and diagnostic outputs.
- Tools to inspect generated logs, telemetry records, diagnostic bundles, crash reports, debug outputs, or equivalent diagnostic artefacts.

4. Assessment activities
The assessment shall at least:

1) Identify the Hypervisor logs, telemetry, and diagnostic outputs produced by the product.
2) Generate representative logs, telemetry, and diagnostic outputs under normal and error conditions.
3) Inspect the generated outputs and verify that they do not include:
- plaintext authentication credentials;
- full cryptographic keys;
- guest workload payload data;
- VM memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data.
4) Where a documented diagnostic function intentionally collects sensitive data, verify that the diagnostic function is explicitly documented and that the collected data is protected according to the product security documentation.

5. Assessment verdict

- Pass: Hypervisor logs, telemetry, and diagnostics do not include plaintext authentication credentials, full cryptographic keys, guest workload payload data, or VM memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data, except where explicitly required for a documented diagnostic function and protected according to the product security documentation.
- Fail: Hypervisor logs, telemetry, or diagnostics include plaintext authentication credentials, full cryptographic keys, guest workload payload data, or VM memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data, and the inclusion is not explicitly required for a documented diagnostic function or is not protected according to the product security documentation.

6. Assessment evidence

- [DESIGN] Documentation of Hypervisor logging, telemetry, and diagnostic data categories.
- [CONFIG] Configuration of Hypervisor logging, telemetry, and diagnostic functions.
- [LOG] / [OBSERVATION] Samples of generated logs, telemetry records, diagnostic bundles, crash reports, debug outputs, or equivalent artefacts showing that prohibited sensitive data is not included.
- [DESIGN] / [CONFIG] Where sensitive diagnostic collection is intentionally supported, documentation and configuration showing the diagnostic purpose and applied protections.

6.3.1.17.2 Assessment Case AC-H-DM-002

1. Assessment reference
Requirement: REQ-H-DM-002 (Basic)

2. Assessment objective
Verify that:

1) The Hypervisor provides administrative controls to restrict and, where necessary, disable collection of specific categories of log, telemetry and diagnostic data.
2) These controls include the ability to disable collection of particular log types, telemetry streams, metrics, or debug payload capture.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing configurable categories of logs, telemetry streams, metrics and any debug/payload capture options.
- Documentation of administrative roles/permissions required to modify logging/telemetry configuration.
- A test configuration where different log categories, telemetry streams and metrics can be enabled/disabled and where events can be generated to exercise them.

4. Assessment activities
The assessment shall at least:

1) Identify, from documentation and configuration, the configurable categories (e.g. audit logs vs debug logs, specific telemetry streams, metrics groups, payload capture options).
2) Using administrative controls, configure a subset of categories to be disabled (e.g. disable a particular debug log type, a telemetry stream and, if supported, debug payload capture).
3) Generate events and workload activity that would normally produce entries in the disabled categories and verify that:

- no new records are produced for the disabled log types/streams/metrics; and
- other, non-disabled categories continue to function as configured.

4) Re-enable one or more previously disabled categories and verify that records are again produced for those categories when new events occur.

5. Assessment verdict

- Pass: administrators can restrict/disable specific log types, telemetry streams, metrics and any debug payload capture; disabled categories no longer produce data while others continue to operate; and re-enabling restores data collection as configured.
- Fail: logging/telemetry categories cannot be selectively restricted or disabled; or documented categories remain collected despite being configured as disabled.

6. Assessment evidence

- [CONFIG] Logging/telemetry/metrics configuration before and after enabling/disabling categories, including debug/payload options.
- [DESIGN] Documentation of administrative controls for data collection and category granularity (log types, streams, metrics, payload capture).
- [LOG] / [OBSERVATION] Evidence showing presence/absence of data from specific categories when enabled vs disabled, demonstrating effective restriction and disabling.

6.3.1.18 Assessment for Decommissioning

6.3.1.18.1 Assessment Case AC-H-DECOM-001

1. Assessment reference
Requirement: REQ-H-DECOM-001 (Basic - Single class)

2. Assessment objective
Verify that the Hypervisor provides a mechanism to decommission a guest virtual machine that renders the associated images, snapshot data, and configuration data managed by the Hypervisor no longer retrievable or usable through the interfaces and functions provided by the Hypervisor.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the Hypervisor decommissioning mechanism and the types of data it affects.
- Documentation identifying the images, snapshot data, and configuration data managed by the Hypervisor for the assessed guest virtual machine.
- A test configuration with at least one guest virtual machine and associated Hypervisor-managed images, snapshot data, and configuration data accessible through Hypervisor interfaces and functions before decommissioning.
- Access to the Hypervisor interfaces and functions normally used to manage or retrieve that data.

4. Assessment activities
The assessment shall at least:

- Review documentation to identify the decommissioning mechanism and the associated images, snapshot data, and configuration data managed by the Hypervisor for the assessed guest virtual machine.
- Confirm, before decommissioning, that the guest virtual machine and its associated Hypervisor-managed images, snapshot data, and configuration data are retrievable or usable through the interfaces and functions provided by the Hypervisor.
- Invoke the documented decommissioning mechanism for the guest virtual machine.
- Verify, after decommissioning, that the guest virtual machine and its associated Hypervisor-managed images, snapshot data, and configuration data are no longer retrievable or usable through the interfaces and functions provided by the Hypervisor.

5. Assessment verdict

- Pass: The Hypervisor provides a decommissioning mechanism and, after its invocation, the assessed guest virtual machine and its associated Hypervisor-managed images, snapshot data, and configuration data are no longer retrievable or usable through the interfaces and functions provided by the Hypervisor.
- Fail: No decommissioning mechanism is provided; or, after decommissioning, the assessed guest virtual machine or any associated Hypervisor-managed images, snapshot data, or configuration data remain retrievable or usable through the interfaces and functions provided by the Hypervisor.

6. Assessment evidence

- [DESIGN] Documentation of the Hypervisor decommissioning mechanism and the associated data types within scope.
- [SCOPE] Identification of the assessed guest virtual machine and the associated Hypervisor-managed images, snapshot data, and configuration data.
- [CONFIG] Relevant Hypervisor configuration for the assessed decommissioning procedure.
- [LOG] / [OBSERVATION] Evidence showing the accessible state before decommissioning and the non-retrievable / non-usable state after decommissioning through the interfaces and functions provided by the Hypervisor.

6.3.2 M&O System

6.3.2.1 Assessment for Authentication

6.3.2.1.1 Assessment Case AC-M&O-AUTH-001

1. Assessment reference
Requirement: REQ-M&O-AUTH-001 (Basic)

2. Assessment objective
Verify that all administrative interfaces of the M&O system require authentication using unique, non-default credentials, and that default credentials are not active in the delivered product.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying all administrative interfaces exposed by the M&O system and the authentication mechanisms applied to each
- Documentation confirming that no default or shared credentials are pre-configured in the delivered product, or that any default credentials are required to be changed before first use
- A deployed M&O system accessible via its administrative interfaces
- Tools to inspect interface authentication configuration and to attempt authentication with default or blank credentials

4. Assessment activities
The assessment shall at least:

- Enumerate all administrative interfaces (e.g. management console, CLI, API endpoints) and confirm that each requires authentication before granting access
- Inspect the delivered configuration and documentation to confirm that no default, shared, or blank credentials are active; or that the product enforces a credential change on first use
- Attempt to access each administrative interface without credentials or using default credentials and verify that access is denied

5. Assessment verdict

- Pass: all administrative interfaces require authentication using unique, non-default credentials; default or blank credentials do not grant access to the delivered product
- Fail: one or more administrative interfaces do not require authentication; or default/blank credentials grant administrative access without requiring a credential change

6. Assessment evidence

- [CONFIG] Interface configuration showing authentication requirements for each administrative interface
- [DESIGN] Documentation confirming the absence of default credentials or the enforced first-use credential change process
- [LOG] / [OBSERVATION] Results of authentication attempts confirming that unauthenticated and default-credential access is denied

6.3.2.1.2 Assessment Case AC-M&O-AUTH-002

1. Assessment reference
Requirement: REQ-M&O-AUTH-002 (Elevated)

2. Assessment objective
Verify that the M&O system supports multi-factor authentication (MFA) for administrative access, enforces revocable authentication mechanisms for administrative and service API access, and authenticates internal component communication.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the MFA mechanism(s) supported for administrative access and the supported second-factor types

- Documentation describing how authentication credentials or sessions for administrative and service API access can be revoked
- Documentation describing how internal component communication within the M&O system is authenticated
- A test environment with MFA enabled for at least one administrative account
- Tools to inspect authentication sessions, test revocation, and examine internal component communication

4. Assessment activities
The assessment shall at least:

- Verify that MFA is supported for administrative access and that at least one administrative account can be configured to require a second factor for authentication
- Attempt to authenticate as an MFA-enabled administrative account using only the primary credential (e.g. password alone) and verify that access is denied without the second factor
- Confirm that authentication credentials or active sessions for administrative and service API access can be revoked, and that revoked credentials/sessions no longer grant access
- Review internal component communication configuration and confirm that authentication is applied between internal M&O components

5. Assessment verdict

- Pass: MFA is supported and enforced for configured administrative accounts; authentication mechanisms for administrative and service API access are revocable; internal component communication is authenticated
- Fail: MFA is not supported or not enforceable for administrative access; authentication mechanisms cannot be revoked; or internal component communication is unauthenticated

6. Assessment evidence

- [CONFIG] MFA configuration for administrative accounts and internal component authentication settings
- [DESIGN] Documentation of MFA mechanisms, revocable authentication, and internal communication authentication
- [LOG] / [OBSERVATION] Evidence that single-factor access is denied for MFA-enabled accounts and that revoked credentials/sessions do not grant access

6.3.2.1.3 Assessment Case AC-M&O-AUTH-003

1. Assessment reference
Requirement: REQ-M&O-AUTH-003 (Advanced)

2. Assessment objective
Verify that the M&O system supports integration with external identity and trust services to validate user and service identities and supports automated credential rotation and revocation where available.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing supported integrations with external identity and trust services (e.g. LDAP, SAML, OIDC, PKI) and how user/service identity validation is performed
- Documentation describing any automated credential rotation and revocation capabilities, and their operational prerequisites
- A test environment with at least one external identity service integrated with the M&O system
- Tools to inspect identity validation and credential lifecycle management behaviour

4. Assessment activities
The assessment shall at least:

- Configure integration with at least one external identity or trust service and verify that user or service authentication is delegated to and validated by that service
- Confirm that identities not recognized by the external service are denied access
- Where automated credential rotation is supported, verify that credentials are rotated according to the configured policy and that rotated credentials take effect without requiring manual intervention
- Where automated revocation is supported, revoke a credential in the external service and verify that the M&O system rejects subsequent access attempts using the revoked credential

5. Assessment verdict

- Pass: the M&O system successfully integrates with an external identity/trust service for identity validation; identities not recognized by the external service are denied access; automated credential rotation and revocation operate as documented where supported
- Fail: integration with external identity services is not supported or does not function as documented; or automated credential rotation/revocation is not supported or does not operate correctly

6. Assessment evidence

- [CONFIG] Integration configuration for external identity/trust services and credential rotation/revocation settings
- [DESIGN] Documentation of external identity integration architecture and automated credential lifecycle management
- [LOG] / [OBSERVATION] Evidence of successful identity validation via the external service and, where applicable, automated rotation and revocation outcomes

6.3.2.2 Assessment for Authorization

6.3.2.2.1 Assessment Case AC-M&O-AUTHZ-001

1. Assessment reference
Requirement: REQ-M&O-AUTHZ-001 (Basic)

2. Assessment objective
Verify that the M&O system restricts all administrative actions to authenticated accounts and enforces a default-deny policy for all resource access and control functions.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the access control model applied to administrative actions and resource access
- A deployed M&O system with at least one configured administrative account
- Tools to attempt unauthenticated and unauthorized resource access and control operations

4. Assessment activities
The assessment shall at least:

- Attempt to perform administrative actions (e.g. configuration changes, lifecycle operations) without authentication and verify that all such attempts are denied
- Confirm that resource access and control functions apply a default-deny policy, i.e. access is denied unless explicitly permitted
- Verify that authenticated accounts can only perform actions explicitly permitted for their assigned role

5. Assessment verdict

- Pass: all administrative actions are restricted to authenticated accounts; unauthenticated access to resource access and control functions is denied; a default-deny policy is enforced
- Fail: unauthenticated accounts can perform administrative actions or access resources; or a default-permit policy is applied to resource access

6. Assessment evidence

- [CONFIG] Access control configuration enforcing authentication requirements and default-deny policy
- [DESIGN] Documentation of the access control model for administrative actions and resource access
- [LOG] / [OBSERVATION] Evidence that unauthenticated and unauthorized access attempts are denied

6.3.2.2.2 Assessment Case AC-M&O-AUTHZ-002

1. Assessment reference
Requirement: REQ-M&O-AUTHZ-002 (Elevated)

2. Assessment objective
Verify that the M&O system enforces granular authorization for administrative users and service accounts, binding permitted actions to authenticated identities and enforcing least-privilege permissions by default. The assessment shall not require a specific authorization model, provided that the implemented model supports granular permissions and least-privilege enforcement. The implemented authorization model may be role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the implemented M&O authorization model, including whether it is role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.
- Documentation identifying protected administrative operations, service-account operations, and management functions.
- Documentation identifying authenticated administrative users, service accounts, permissions, policy rules, or equivalent authorization bindings.
- A test configuration with at least two authenticated identities or service accounts having different permissions for representative management operations.
- Tools or interfaces to attempt permitted and non-permitted management operations using those authenticated identities or service accounts.

4. Assessment activities
The assessment shall at least:

- Identify the M&O administrative, service-account, and management operations covered by REQ-M&O-AUTHZ-002.
- Review the implemented authorization model and confirm that permitted actions are bound to authenticated identities or service accounts.
- Verify that least-privilege permissions are enforced by default.
- Attempt a permitted management operation using an authenticated identity or service account authorized for that action and verify that the action succeeds.
- Attempt the same management operation using an authenticated identity or service account not authorized for that action and verify that the action is denied.
- Verify that the evidence demonstrates granular authorization without requiring a specific authorization model.

5. Assessment verdict

- Pass: the M&O system enforces granular authorization for administrative users and service accounts, binds permitted actions to authenticated identities, and enforces least-privilege permissions by default, regardless of whether the implemented model is role-based, attribute-based, policy-based, capability-based, or equivalent.
- Fail: the M&O system does not enforce granular authorization; or permitted actions are not bound to authenticated identities or service accounts; or least-privilege permissions are not enforced by default; or an authenticated identity or service account can perform an action for which it is not authorized.

6. Assessment evidence

- [DESIGN] Documentation of the implemented M&O authorization model and its binding of permissions to authenticated identities or service accounts.
- [CONFIG] Authorization policy, role, attribute, rule, capability, or equivalent configuration showing permitted and non-permitted actions.
- [LOG] / [OBSERVATION] Evidence of permitted actions succeeding for authorized identities or service accounts and non-permitted actions being denied for unauthorized identities or service accounts.

6.3.2.2.3 Assessment Case AC-M&O-AUTHZ-003

1. Assessment reference
Requirement: REQ-M&O-AUTHZ-003 (Advanced)

2. Assessment objective
Verify that the M&O system enforces fine-grained, policy-driven, or context-aware authorization for administrative users and service accounts, including separation of duties, contextual policy evaluation, or integration with external identity, access-management, or policy decision systems where such integration is part of the declared product configuration. Audit events for privilege changes and authorization decisions are assessed under the logging requirements and shall not be introduced as an additional obligation in this authorization assessment case.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the M&O fine-grained, policy-driven, or context-aware authorization model.
- Documentation identifying separation-of-duties rules, contextual policy inputs, policy decision points, or equivalent mechanisms supported by the M&O system, where applicable.
- Documentation identifying whether integration with external identity, access-management, or policy decision systems is part of the declared product configuration.
- A test configuration with representative administrative users, service accounts, permissions, contextual attributes, policy inputs, or external authorization integration, as applicable to the declared product configuration.

4. Assessment activities
The assessment shall at least:

- Review the documented fine-grained, policy-driven, or context-aware authorization model.
- Verify that the M&O system enforces the applicable authorization policy for representative administrative and service-account operations.
- Where separation of duties is supported or required by the declared configuration, verify that conflicting or incompatible operations are not permitted to the same identity unless explicitly authorized by policy.
- Where context-aware enforcement is supported or applicable, verify that authorization decisions take the documented contextual inputs into account.
- Where integration with external identity, access-management, or policy decision systems is part of the declared product configuration, verify that authorization decisions use the integrated system as documented.

5. Assessment verdict

- Pass: the M&O system enforces fine-grained, policy-driven, or context-aware authorization for administrative users and service accounts, including separation of duties, contextual policy evaluation, or external authorization integration where applicable to the requirement and declared product configuration.
- Fail: the M&O system does not enforce fine-grained, policy-driven, or context-aware authorization where required; or separation-of-duties, contextual policy evaluation, or external authorization integration is claimed but not implemented or not effective in the declared product configuration.

6. Assessment evidence

- [DESIGN] Documentation of the fine-grained, policy-driven, or context-aware M&O authorization model.
- [CONFIG] Authorization policy, contextual policy, separation-of-duties configuration, or external authorization integration configuration, as applicable.
- [LOG] / [OBSERVATION] Evidence that authorization decisions are enforced according to the documented policy and declared product configuration.

6.3.2.3 Assessment for Secure Configuration

6.3.2.3.1 Assessment Case AC-M&O-CFG-001

1. Assessment reference
Requirement: REQ-M&O-CFG-001 (Basic)

2. Assessment objective
Verify that the M&O system is delivered with secure default settings, including least-privilege access roles, disabled unused services or ports, and enforced secure communication protocols.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the default configuration of the M&O system as delivered, including default roles, enabled services/ports, and communication protocol settings
- A freshly deployed M&O system in its default configuration (or documentation of the default state)
- Tools to inspect exposed services, open ports, and communication protocol settings

4. Assessment activities
The assessment shall at least:

- Inspect the default configuration and confirm that access roles are defined with least-privilege assignments and that no overly broad default roles are active
- Enumerate exposed services and open ports in the default configuration and verify that unused or unnecessary services and ports are disabled by default
- Confirm that secure communication protocols are enforced by default for administrative and management interfaces

5. Assessment verdict

- Pass: the M&O system is delivered with least-privilege default roles, unused services and ports disabled, and secure communication protocols enforced in the default configuration
- Fail: the default configuration includes overly broad roles, unnecessary services or ports enabled, or insecure communication protocols active by default

6. Assessment evidence

- [CONFIG] Default configuration state showing role assignments, enabled services/ports, and communication protocol settings
- [DESIGN] Documentation of secure defaults and hardening guidance
- [OBSERVATION] Results of service/port enumeration and protocol inspection on the default configuration

6.3.2.3.2 Assessment Case AC-M&O-CFG-002

1. Assessment reference
Requirement: REQ-M&O-CFG-002 (Elevated)

2. Assessment objective
Verify that the M&O system requires explicit authorization before applying configuration changes to the hypervisor or managed workloads, and before applying changes to its own internal configuration.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the authorization controls applied before configuration changes are permitted, both for managed hypervisors/workloads and for the M&O system itself
- A test environment with accounts of varying privilege levels
- Tools to attempt configuration changes without authorization or using under-privileged accounts

4. Assessment activities
The assessment shall at least:

- Attempt to apply a configuration change to a managed hypervisor or workload using an account without the required authorization and verify that the change is denied
- Attempt to apply a change to the M&O system's own internal configuration without the required authorization and verify that the change is denied
- Confirm that authorized configuration changes are permitted and take effect correctly
- Review audit logs to confirm that authorized and denied configuration change attempts are recorded

5. Assessment verdict

- Pass: configuration changes to managed hypervisors/workloads and to the M&O system's own configuration require explicit authorization; unauthorized change attempts are denied and logged
- Fail: configuration changes to managed targets or to the M&O system's own configuration can be applied without explicit authorization

6. Assessment evidence

- [CONFIG] Authorization settings for configuration change operations
- [DESIGN] Documentation of the authorization model for configuration changes
- [LOG] / [OBSERVATION] Audit log entries showing authorized and denied configuration change attempts

6.3.2.3.3 Assessment Case AC-M&O-CFG-003

1. Assessment reference
Requirement: REQ-M&O-CFG-003 (Advanced)

2. Assessment objective
Verify that the M&O system enforces security configuration baselines for all managed hypervisors and its own components and automatically detects and alerts on deviations from these baselines.

3. Assessment preparation
The assessment shall have access to:

- Documentation of the security configuration baselines defined by the manufacturer for managed hypervisors and for M&O system components
- Documentation describing the deviation detection and alerting mechanism
- A test environment with the baseline configuration applied to at least one managed hypervisor and the M&O system
- Tools to introduce a deliberate configuration deviation and to observe detection and alerting behaviour

4. Assessment activities
The assessment shall at least:

- Review the documented security configuration baselines and confirm they are applied to the managed hypervisor(s) and M&O system components in the test environment
- Introduce a deliberate deviation from the security baseline on a managed hypervisor or M&O component (e.g. change a security setting, disable a required service)
- Verify that the M&O system detects the deviation automatically and generates an alert or notification
- Confirm that the alert contains sufficient information to identify the affected component and the nature of the deviation

5. Assessment verdict

- Pass: security configuration baselines are enforced for managed hypervisors and M&O components; deliberate deviations are automatically detected and an alert is generated with sufficient detail to identify the component and deviation
- Fail: security configuration baselines are not defined or enforced; deviations are not automatically detected; or alerts are not generated or are insufficient to identify the affected component and deviation

6. Assessment evidence

- [CONFIG] Baseline configuration applied to managed hypervisors and M&O components
- [DESIGN] Documentation of baseline definitions, deviation detection mechanisms, and alerting procedures
- [LOG] / [OBSERVATION] Alert or notification generated in response to the introduced configuration deviation

6.3.2.4 Assessment for Communication Security

6.3.2.4.1 Assessment Case AC-M&O-COM-001

1. Assessment reference
Requirement: REQ-M&O-COM-001 (Basic)

2. Assessment objective
Verify that the M&O system protects the confidentiality and integrity of management and orchestration communications exchanged through M&O interfaces and communication paths used to control or manage hypervisors.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying M&O interfaces and communication paths used to control or manage hypervisors
- Documentation describing the confidentiality and integrity protection mechanisms applied to those interfaces and paths
- A test environment with the M&O system communicating with at least one managed hypervisor

- Network traffic capture tools to inspect M&O communications

4. Assessment activities
The assessment shall at least:

- Enumerate the M&O interfaces and communication paths used to manage hypervisors in the assessed configuration
- Capture network traffic on at least one M&O communication path and verify that management and orchestration data is protected by a confidentiality and integrity mechanism (e.g. TLS)
- Confirm that management traffic is not transmitted in plaintext over network paths accessible to unauthorized parties

5. Assessment verdict

- Pass: management and orchestration communications on M&O interfaces and paths used to control/manage hypervisors are protected for confidentiality and integrity; no plaintext management traffic is transmitted over accessible network paths
- Fail: management or orchestration communications are transmitted without confidentiality or integrity protection; or plaintext management traffic is observable on network paths

6. Assessment evidence

- [CONFIG] Communication security settings for M&O interfaces and management paths
- [DESIGN] Documentation of communication security architecture for M&O-to-hypervisor paths
- [OBSERVATION] Network capture or inspection results confirming encrypted management communications

6.3.2.4.2 Assessment Case AC-M&O-COM-002

1. Assessment reference
Requirement: REQ-M&O-COM-002 (Elevated)

2. Assessment objective
Verify that the M&O system enforces certificate-based mutual authentication for communications used to exchange management or orchestration data with managed hypervisors.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the mutual authentication mechanism and the certificate infrastructure used for M&O-to-hypervisor communications
- A test environment with mutual authentication configured between the M&O system and at least one managed hypervisor
- Tools to inspect certificate validation and to attempt connections without valid certificates

4. Assessment activities
The assessment shall at least:

- Confirm that mutual authentication is configured and active for M&O-to-hypervisor management communications
- Attempt to establish a management connection from the M&O system to a managed hypervisor using an invalid, expired, or missing certificate and verify that the connection is rejected
- Attempt to connect to the M&O system from a simulated hypervisor using an invalid certificate and verify that the connection is rejected

5. Assessment verdict

- Pass: certificate-based mutual authentication is enforced for M&O-to-hypervisor management communications; connections without valid certificates are rejected in both directions
- Fail: mutual authentication is not enforced; connections with invalid or missing certificates are accepted; or certificate validation can be bypassed

6. Assessment evidence

- [CONFIG] Certificate and mutual authentication configuration for M&O-to-hypervisor communications
- [DESIGN] Documentation of the certificate infrastructure and mutual authentication enforcement
- [LOG] / [OBSERVATION] Evidence that invalid-certificate connection attempts are rejected

6.3.2.4.3 Assessment Case AC-M&O-COM-003

1. Assessment reference
Requirement: REQ-M&O-COM-003 (Advanced)

2. Assessment objective
Verify that the M&O system ensures control operations are performed only over cryptographically protected and mutually authenticated channels, and only toward managed hypervisors whose identity has been successfully verified and authorized.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing how hypervisor identity verification and authorization are performed before control operations are permitted
- Documentation describing the cryptographic protection and mutual authentication applied to control operation channels
- A test environment with the M&O system managing at least one hypervisor via a verified and authorized channel
- Tools to attempt control operations toward a hypervisor whose identity has not been verified or authorized

4. Assessment activities
The assessment shall at least:

- Confirm that control operations are performed only over channels protected by cryptographic confidentiality and integrity mechanisms and mutually authenticated
- Attempt to direct a control operation toward a managed hypervisor using an unverified or unauthorized hypervisor identity and verify that the operation is rejected
- Attempt to send a control operation over an unprotected channel (e.g. without TLS) and verify that it is not accepted

5. Assessment verdict

- Pass: control operations are performed only over cryptographically protected, mutually authenticated channels; operations directed at unverified or unauthorized hypervisor identities are rejected; operations over unprotected channels are rejected
- Fail: control operations can be performed over unprotected channels; or operations toward unverified/unauthorized hypervisor identities are accepted

6. Assessment evidence

- [CONFIG] Channel protection and identity verification/authorization configuration for control operations

- [DESIGN] Documentation of the control operation channel security model and hypervisor identity verification procedure
- [LOG] / [OBSERVATION] Evidence that control operations over unprotected channels or toward unverified identities are rejected

6.3.2.5 Assessment for Integrity Protection

6.3.2.5.1 Assessment Case AC-M&O-INT-001

1. Assessment reference
Requirement: REQ-M&O-INT-001 (Basic)

2. Assessment objective
Verify that the M&O system validates the integrity of orchestration artefacts, policies, templates, scripts, or configuration bundles before execution or application, using a trusted reference value, such as a cryptographic digest, digital signature, or equivalent integrity reference.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the integrity verification mechanism used for orchestration artefacts, policies, templates, scripts, or configuration bundles.
- Documentation identifying the trusted reference values used for integrity verification, such as cryptographic digests, digital signatures, or equivalent integrity references.
- A test environment with representative orchestration artefacts, policies, templates, scripts, or configuration bundles.
- Test artefacts including:
  - one valid artefact with a matching trusted reference value;
  - one modified artefact with a mismatched or invalid trusted reference value;
  - one artefact without the required integrity reference, where applicable to the configured policy.

4. Assessment activities
The assessment shall at least:

1) Identify the orchestration artefacts, policies, templates, scripts, or configuration bundles covered by REQ-M&O-INT-001.
2) Review the integrity verification mechanism and confirm that it uses trusted reference values, such as cryptographic digests, digital signatures, or equivalent integrity references.
3) Present a valid artefact with a matching trusted reference value and verify that it is accepted.
4) Present a modified artefact with a mismatched or invalid trusted reference value and verify that it is rejected and not executed or applied.
5) Where artefacts without required integrity references are not permitted by policy, present such an artefact and verify that it is rejected.
6) For pull-based reconciliation flows, where supported, verify that integrity verification is performed at the reconciliation point before state is applied to managed targets.

5. Assessment verdict

- Pass: orchestration artefacts, policies, templates, scripts, or configuration bundles with valid trusted reference values are accepted; artefacts with invalid or mismatched integrity references are rejected and not executed or applied; integrity verification is applied at import, before execution, before application, or at the reconciliation point as documented; and the mechanism supports cryptographic digests, digital signatures, or equivalent integrity references according to the configured policy.

- Fail: artefacts with invalid or mismatched integrity references are accepted and executed or applied; integrity verification is not applied before execution, application, or reconciliation as documented; or the mechanism is limited to an implementation-specific integrity reference and does not support the integrity reference type required by the documented policy.

6. Assessment evidence

- [CONFIG] Integrity verification configuration for orchestration artefact import, reconciliation, execution, or application.
- [DESIGN] Documentation of the integrity verification mechanism, covered artefact types, trusted reference values, and any pull-based reconciliation verification behaviour.
- [LOG] / [OBSERVATION] Results showing acceptance of valid artefacts and rejection of integrity-failed artefacts, including in pull-based reconciliation flows where supported.

6.3.2.5.2 Assessment Case AC-M&O-INT-002

1. Assessment reference
Requirement: REQ-M&O-INT-002 (Elevated)

2. Assessment objective
Verify that the M&O system verifies the integrity and authenticity of orchestration artefacts using digital signatures validated against trusted keys, certificates, or equivalent trust anchors before execution or application.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the digital signature verification mechanism used for orchestration artefacts.
- Documentation identifying the trusted keys, certificates, or equivalent trust anchors used to validate orchestration artefacts.
- A test environment with the M&O system configured to validate artefact signatures.
- Test artefacts including:
  - one artefact signed using a trusted key, certificate, or equivalent trust anchor;
  - one artefact signed using an untrusted key, certificate, or trust anchor;
  - one unsigned artefact.

4. Assessment activities
The assessment shall at least:

- Review the digital signature verification mechanism and confirm that signature validation is performed using trusted keys, certificates, or equivalent trust anchors.
- Present a correctly signed artefact validated by a trusted key, certificate, or equivalent trust anchor and verify that the M&O system accepts and executes or applies it.
- Present an artefact signed with an untrusted key, certificate, or trust anchor and verify that the M&O system rejects it before execution or application.
- Present an unsigned artefact and verify that the M&O system rejects it where signature validation is required by the configured policy.
- Confirm that signature validation is performed before execution or application.

5. Assessment verdict

- Pass: artefacts signed using trusted keys, certificates, or equivalent trust anchors are accepted; artefacts signed using untrusted keys, certificates, or trust anchors are rejected before execution or application; unsigned artefacts are rejected where signature validation is required by policy; and signature validation is performed before execution or application.
- Fail: artefacts signed using untrusted keys, certificates, or trust anchors are accepted; or unsigned artefacts are accepted where signature validation is required by policy; or signature validation is not performed before execution or application.

6. Assessment evidence

- [CONFIG] Configuration of trusted keys, certificates, or equivalent trust anchors for orchestration artefact signature validation.
- [DESIGN] Documentation of the digital signature verification architecture.
- [LOG] / [OBSERVATION] Test results showing acceptance of trusted signed artefacts and rejection of untrusted or unsigned artefacts, where applicable.

6.3.2.5.3 Assessment Case AC-M&O-INT-003

1. Assessment reference
Requirement: REQ-M&O-INT-003 (Advanced)

2. Assessment objective
Verify that the M&O system validates the integrity and authenticity of all third-party plugins, extensions, or integration modules before activation, and supports administrative functions to register and revoke trusted signing keys.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the integrity and authenticity validation mechanism for third-party plugins, extensions, and integration modules
- Documentation describing the administrative functions for registering and revoking trusted signing keys
- A test environment with at least one third-party plugin or extension available for installation or activation
- Test plugins: one signed with a trusted key, one unsigned or signed with an untrusted key

4. Assessment activities
The assessment shall at least:

- Attempt to activate a plugin or extension signed with a trusted key and verify that it is accepted
- Attempt to activate an unsigned or untrusted-key-signed plugin and verify that activation is rejected
- Use the administrative interface to register a new signing key and verify that artefacts signed with that key are subsequently accepted
- Use the administrative interface to revoke a signing key and verify that artefacts signed with the revoked key are subsequently rejected

5. Assessment verdict

- Pass: plugins/extensions signed with trusted keys are accepted; untrusted or unsigned plugins are rejected; administrative registration and revocation of signing keys function correctly and take effect on subsequent validation
- Fail: untrusted or unsigned plugins can be activated; or administrative key registration/revocation is not supported or does not take effect on validation

6. Assessment evidence

- [CONFIG] Plugin/extension trust configuration including registered and revoked signing keys
- [DESIGN] Documentation of plugin integrity validation and key management architecture
- [LOG] / [OBSERVATION] Evidence of accepted trusted-key-signed plugins, rejected untrusted plugins, and effective key registration and revocation

6.3.2.6 Assessment for Logging

6.3.2.6.1 Assessment Case AC-M&O-LOG-001

1. Assessment reference
Requirement: REQ-M&O-LOG-001 (Basic)

2. Assessment objective
Verify that the M&O system logs all administrative actions, including user logins, configuration changes, VM lifecycle events, orchestration workflows, and security-related alerts.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying the administrative actions, lifecycle events, orchestration workflows, and security alerts that are logged by the M&O system
- A test environment with audit logging enabled
- Tools to perform representative administrative actions and to inspect the resulting audit log entries

4. Assessment activities
The assessment shall at least:

- Perform a user login (successful and failed) and verify that both are recorded in the audit log
- Apply a configuration change and verify that the change is recorded
- Execute a VM lifecycle event (e.g. start, stop, or migration) and verify that it is recorded
- Execute an orchestration workflow and verify that it is recorded
- Trigger a security-related alert (e.g. authentication failure, policy violation) and verify that it is recorded

5. Assessment verdict

- Pass: user logins, configuration changes, VM lifecycle events, orchestration workflows, and security-related alerts are all recorded in the audit log
- Fail: one or more required event categories (logins, configuration changes, VM lifecycle events, orchestration workflows, or security alerts) are not recorded in the audit log

6. Assessment evidence

- [CONFIG] Audit logging configuration for the M&O system
- [DESIGN] Documentation of the event categories covered by audit logging
- [LOG] Audit log entries demonstrating recording of each required event category

6.3.2.6.2 Assessment Case AC-M&O-LOG-002

1. Assessment reference
Requirement: REQ-M&O-LOG-002 (Elevated)

2. Assessment objective
Verify that the M&O system protects stored logs against tampering and supports secure export of logs to external logging or SIEM systems using encrypted, authenticated protocols.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the mechanism used to protect stored audit logs against unauthorized modification or deletion
- Documentation describing the secure export mechanism and the protocol(s) used to transmit logs to external systems
- A test environment with log protection enabled and, where available, an external logging or SIEM target configured
- Tools to attempt unauthorized log modification and to inspect log export traffic

4. Assessment activities
The assessment shall at least:

- Attempt to modify or delete audit log entries without appropriate authorization and verify that the attempt is prevented or detected
- Confirm that log export to an external logging or SIEM system uses an encrypted and authenticated protocol
- Verify that the log export configuration can be set up and that exported logs are received at the external target

5. Assessment verdict

- Pass: stored audit logs are protected against unauthorized modification or deletion; log export to external systems uses encrypted and authenticated protocols
- Fail: audit logs can be modified or deleted without authorization; or log export to external systems is performed over unencrypted or unauthenticated channels

6. Assessment evidence

- [CONFIG] Log protection and log export configuration including protocol and authentication settings
- [DESIGN] Documentation of log tamper-protection mechanisms and secure export architecture
- [LOG] / [OBSERVATION] Evidence of denied unauthorized log modification and encrypted/authenticated log export traffic

6.3.2.6.3 Assessment Case AC-M&O-LOG-003

1. Assessment reference
Requirement: REQ-M&O-LOG-003 (Advanced)

2. Assessment objective
Verify that the M&O system supports cryptographic signing and time-stamping of audit logs to ensure verification of origin and integrity during forensic analysis.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the cryptographic signing and time-stamping mechanisms applied to audit logs
- Documentation describing the verification procedure for checking log signatures and time-stamps
- A test environment with cryptographic log signing and time-stamping enabled
- Tools to verify log signatures and time-stamps, and to detect tampering

4. Assessment activities
The assessment shall at least:

- Collect sample audit log entries and verify that each entry or log batch carries a valid cryptographic signature and time-stamp
- Verify the signature and time-stamp of at least one log entry using the documented verification procedure and confirm the result is valid
- Modify a log entry and attempt to verify the signature and confirm that verification fails, demonstrating tamper detection

5. Assessment verdict

- Pass: audit log entries carry cryptographic signatures and time-stamps; signature and time-stamp verification succeeds for unmodified entries; verification fails for modified entries, demonstrating tamper detection
- Fail: audit log entries do not carry cryptographic signatures or time-stamps; or the signing/time-stamping mechanism cannot be used to detect tampering

6. Assessment evidence

- [CONFIG] Cryptographic signing and time-stamping configuration for audit logs
- [DESIGN] Documentation of log signing architecture, time-stamp sources, and verification procedures
- [LOG] / [OBSERVATION] Sample signed and time-stamped log entries; verification outputs for unmodified and tampered entries

6.3.2.7 Assessment for Secure Update

6.3.2.7.1 Assessment Case AC-M&O-UPD-001

1. Assessment reference
Requirement: REQ-M&O-UPD-001 (Basic)

2. Assessment objective
Verify that:

- The M&O system is delivered in a form that supports authenticity and integrity verification of M&O system updates before installation or activation.
- The M&O system supports authenticity and integrity verification of M&O system updates using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Where the M&O system includes the mechanism that verifies, installs, or activates its updates, this mechanism prevents installation or activation of an update when authenticity or integrity verification fails and records a security-relevant event.
- Where update delivery, verification, installation, or activation is performed by a Host OS update service, package manager, platform update service, appliance update mechanism, external update service, orchestration-based update pipeline, or other Operational Environment mechanism outside the declared M&O system product or outside the responsibility of the manufacturer, the product security documentation identifies that mechanism, the expected security properties, and the assumptions required for secure update of the M&O system.
- Where update installation is delegated to an external system that is part of the declared product, the M&O system verifies and records update provenance and records update installation state for updates applied to itself or orchestrated by it for managed Hypervisors, based on information exposed by that external system.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying the M&O system components delivered as part of the declared M&O system product.
- Documentation describing the update artefacts applicable to those M&O system components.
- Documentation describing how authenticity and integrity verification of M&O system updates is supported before installation or activation, including the verification metadata used, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Documentation identifying whether update delivery, verification, installation, or activation is performed by the M&O system, by another component under the responsibility of the manufacturer, by an external system that is part of the declared product, or by an Operational Environment mechanism.
- Where the M&O system includes the mechanism that verifies, installs, or activates updates, a test configuration allowing use of a valid update artefact and an update artefact with invalid, missing, revoked, or intentionally modified authenticity or integrity protection, or an equivalent test condition.
- Where update installation is delegated to an external system that is part of the declared product, documentation describing how the M&O system verifies and records update provenance and installation state.
- Where update delivery, verification, installation, or activation is performed by an Operational Environment mechanism, product security documentation identifying the expected update mechanism, expected security properties, and assumptions required for secure update of the M&O system.

4. Assessment activities
The assessment shall at least:

1) Identify the M&O system components and update artefacts covered by REQ-M&O-UPD-001.
2) Check that the M&O system is delivered in a form that supports authenticity and integrity verification of M&O system updates before installation or activation.
3) Check that the update artefacts support authenticity and integrity verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
4) Check whether update delivery, verification, installation, or activation is performed by the M&O system, by another component under the responsibility of the manufacturer, by an external system that is part of the declared product, or by an Operational Environment mechanism.
5) Where the M&O system includes the mechanism that verifies, installs, or activates updates, attempt to install or activate an update whose authenticity or integrity verification fails and verify that installation or activation is prevented and that a security-relevant event is recorded.
6) Where update installation is delegated to an external system that is part of the declared product, verify that the M&O system records update provenance and update installation state for updates applied to itself or orchestrated by it for managed Hypervisors, based on information exposed by that external system.
7) Where update delivery, verification, installation, or activation is performed outside the declared M&O system product or outside the responsibility of the manufacturer, verify that the product security documentation identifies the Operational Environment dependency, the expected security properties, and the assumptions required for secure update of the M&O system.

5. Assessment verdict

- Pass: the M&O system is delivered in a form that supports authenticity and integrity verification of M&O system updates before installation or activation; update artefacts support verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata; and, where the M&O system includes the mechanism that verifies, installs, or activates updates, failed authenticity or integrity verification prevents installation or activation and is recorded as a security-relevant event. Where update installation is delegated to an external system that is part of the declared product, the M&O system verifies and records update provenance and records update installation state.

Where update delivery, verification, installation, or activation is performed outside the declared M&O system product or outside the responsibility of the manufacturer, the Operational Environment dependency, expected security properties, and assumptions are documented.

- Fail: the M&O system is not delivered in a form that supports authenticity and integrity verification of M&O system updates before installation or activation; or the update artefacts do not support the required verification metadata; or, where the M&O system includes the mechanism that verifies, installs, or activates updates, an update whose authenticity or integrity verification fails can be installed or activated, or the failure is not recorded as a security-relevant event; or, where update installation is delegated to an external system that is part of the declared product, the M&O system does not verify and record update provenance or does not record update installation state; or the relevant Operational Environment dependency is not documented where update delivery, verification, installation, or activation is performed outside the declared M&O system product or outside the responsibility of the manufacturer.

6. Assessment evidence

- [DESIGN] Documentation of the M&O system update model, including update artefacts and the component or mechanism responsible for update delivery, verification, installation, or activation.
- [DESIGN] Documentation of authenticity and integrity verification metadata used for M&O system updates, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- [CONFIG] Configuration of the M&O system update mechanism, where the M&O system includes the mechanism that verifies, installs, or activates updates.
- [LOG] / [OBSERVATION] Where the M&O system includes the mechanism that verifies, installs, or activates updates, evidence that a valid update is accepted and that an update with failed authenticity or integrity verification is prevented from installation or activation and recorded as a security-relevant event.
- [LOG] / [OBSERVATION] Where update installation is delegated to an external system that is part of the declared product, evidence that update provenance and update installation state are recorded by the M&O system.
- [GUIDANCE] / [OE] Where update delivery, verification, installation, or activation is performed outside the declared M&O system product or outside the responsibility of the manufacturer, product security documentation identifying the Operational Environment dependency, expected security properties, and assumptions required for secure update of the M&O system.

6.3.2.7.2 Assessment Case AC-M&O-UPD-002

1. Assessment reference
Requirement: REQ-M&O-UPD-002 (Elevated)

2. Assessment objective
Verify that:

- The M&O system supports rollback protection for M&O system updates where rollback protection is provided by the M&O system or by an update mechanism under the responsibility of the manufacturer.
- Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed M&O system updates is prevented.
- Authorized rollback to a previously trusted M&O system version is permitted only when:

  a) the rollback image or package is verified for integrity and authenticity;

  b) the rollback action is explicitly authorized by an administrator with appropriate privileges or an equivalent authorized control function; and

  c) the rollback operation is recorded as a security-relevant event.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying whether rollback protection for M&O system updates is provided by the M&O system or by an update mechanism under the responsibility of the manufacturer.
- Documentation describing the rollback protection mechanism, where rollback protection is provided.
- Documentation describing procedures for authorized rollback to a previously trusted M&O system version, where authorized rollback is supported.
- Documentation identifying the administrator privileges or equivalent authorized control function required to authorize rollback, where authorized rollback is supported.
- Documentation describing how rollback operations are recorded as security-relevant events, where rollback is supported.
- A test or demonstration configuration allowing an outdated, revoked, or replayed M&O system update to be attempted, where rollback protection is provided by the M&O system or by an update mechanism under the responsibility of the manufacturer.
- A test or demonstration configuration allowing authorized rollback to a previously trusted M&O system version, where authorized rollback is supported.

4. Assessment activities
The assessment shall at least:

1) Check whether rollback protection for M&O system updates is provided by the M&O system or by an update mechanism under the responsibility of the manufacturer.

2) Where rollback protection is not provided by the M&O system or by an update mechanism under the responsibility of the manufacturer, verify that the product security documentation identifies the update and rollback assumptions applicable to the declared product configuration.

3) Where rollback protection is provided, attempt to install an outdated, revoked, or replayed M&O system update outside the documented authorized rollback procedure and verify that installation is prevented.

4) Where authorized rollback is supported, perform an authorized rollback using the documented procedure and verify that:

a) the rollback image or package is verified for integrity and authenticity;

b) the rollback action requires explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; and

c) the rollback operation is recorded as a security-relevant event.

5. Assessment verdict

- Pass: where rollback protection is provided by the M&O system or by an update mechanism under the responsibility of the manufacturer, unauthorized installation of outdated, revoked, or replayed M&O system updates is prevented; and authorized rollback to a previously trusted M&O system version is permitted only following integrity and authenticity verification of the rollback image or package, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function, and recording of the rollback operation as a security-relevant event. Where rollback protection is not provided by the M&O system or by an update mechanism under the responsibility of the manufacturer, the update and rollback assumptions applicable to the declared product configuration are documented.
- Fail: where rollback protection is provided by the M&O system or by an update mechanism under the responsibility of the manufacturer, outdated, revoked, or replayed M&O system updates can be installed without authorization; or authorized rollback can be performed without integrity and authenticity verification; or rollback can be performed without explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; or the rollback operation is not recorded as a security-relevant event.

6. Assessment evidence

- [DESIGN] Documentation identifying whether rollback protection is provided by the M&O system or by an update mechanism under the responsibility of the manufacturer.
- [DESIGN] Documentation of rollback protection mechanisms and authorized rollback procedures, where rollback protection or authorized rollback is supported.
- [CONFIG] Configuration or policy information showing rollback authorization controls, where authorized rollback is supported.
- [LOG] / [OBSERVATION] Where rollback protection is provided, evidence that outdated, revoked, or replayed updates are prevented outside the authorized rollback procedure.
- [LOG] / [OBSERVATION] Where authorized rollback is supported, evidence that rollback is performed only after integrity and authenticity verification, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function, and recording as a security-relevant event.
- [GUIDANCE] / [OE] Where rollback protection is not provided by the M&O system or by an update mechanism under the responsibility of the manufacturer, documentation identifying update and rollback assumptions applicable to the declared product configuration.

6.3.2.7.3 Assessment Case AC-M&O-UPD-003

1. Assessment reference
Requirement: REQ-M&O-UPD-003 (Advanced)

2. Assessment objective
Verify that where the M&O system orchestrates update actions across managed targets as part of the declared product, it provides, for each managed target, identification of the update action, verification of update outcome, and detection of failed, partial, or inconsistent rollout states.

3. Assessment preparation
The assessment shall have access to:

- Product security documentation identifying whether update orchestration across managed targets is supported as part of the declared product.
- Documentation describing how the M&O system executes update actions across managed targets, including how each update action is identified, outcomes are verified, and failed, partial, or inconsistent rollout states are detected, where such update orchestration is supported.
- A test environment with the M&O system managing at least two targets, only where update orchestration across managed targets is supported as part of the declared product.
- Tools to monitor the multi-target update process and to inspect per-target update outcome records, where such update orchestration is supported.

4. Assessment activities
The assessment shall at least:

1) Check whether update orchestration across managed targets is supported as part of the declared product.
2) If update orchestration across managed targets is not supported as part of the declared product, check that this is identified in the product security documentation and proceed to the verdict based on the condition in REQ-M&O-UPD-003 not being triggered.
3) Where update orchestration across managed targets is supported, initiate an update action across at least two managed targets and observe execution.
4) Confirm that each managed target's update action is individually identified in the M&O system records.
5) Confirm that the outcome of the update action on each managed target is verified and recorded.

6) Simulate a failed, partial, or inconsistent rollout state on at least one managed target and verify that the M&O system detects and records the state.

5. Assessment verdict

- Pass: update orchestration across managed targets is not supported as part of the declared product and this is documented; or, where update orchestration across managed targets is supported, the M&O system identifies update actions per managed target, verifies update outcomes per managed target, and detects failed, partial, or inconsistent rollout states.
- Fail: the product security documentation does not identify whether update orchestration across managed targets is supported as part of the declared product; or update orchestration across managed targets is supported as part of the declared product, but update actions are not identified per managed target; or outcomes are not verified per managed target; or failed, partial, or inconsistent rollout states are not detected.

6. Assessment evidence

- [DESIGN] Product security documentation identifying whether update orchestration across managed targets is supported as part of the declared product.
- [DESIGN] Documentation of per-target update execution, outcome verification, and failure, partial rollout, or inconsistent rollout detection, where update orchestration is supported.
- [CONFIG] Multi-target update configuration and outcome verification settings, where update orchestration is supported.
- [LOG] / [OBSERVATION] Per-target update records showing update-action identification, outcome verification, and detection of failed, partial, or inconsistent rollout states, where update orchestration is supported.

6.3.2.8 Assessment for Decommissioning and Secure Transfer

6.3.2.8.1 Assessment Case AC-M&O-DECOM-001

1. Assessment reference
Requirement: REQ-M&O-DECOM-001 (Basic - Single class)

2. Assessment objective
Verify that the M&O system provides a mechanism to remove orchestration data, management configuration, and associated metadata managed by the M&O system such that this data is no longer retrievable or usable through the interfaces and functions provided by the M&O system.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the M&O removal mechanism and the orchestration data, management configuration, and associated metadata within its scope.
- A test configuration containing M&O-managed orchestration data, management configuration, and associated metadata accessible through M&O interfaces and functions before removal.
- Access to the M&O interfaces and functions normally used to manage or retrieve that data.

4. Assessment activities
The assessment shall at least:

1) Review documentation to identify the removal mechanism and the orchestration data, management configuration, and associated metadata managed by the M&O system for the assessment.
2) Confirm, before removal, that the assessed data is retrievable or usable through the interfaces and functions provided by the M&O system.
3) Invoke the documented removal mechanism.

4) Verify, after removal, that the assessed data is no longer retrievable or usable through the interfaces and functions provided by the M&O system.

5. Assessment verdict

- Pass: The M&O system provides a removal mechanism and, after its invocation, the assessed orchestration data, management configuration, and associated metadata are no longer retrievable or usable through the interfaces and functions provided by the M&O system.
- Fail: No removal mechanism is provided; or, after removal, the assessed data remains retrievable or usable through the interfaces and functions provided by the M&O system.

6. Assessment evidence

- [DESIGN] Documentation of the M&O removal mechanism and associated data within scope.
- [SCOPE] Identification of the assessed orchestration data, management configuration, and associated metadata.
- [CONFIG] Relevant M&O configuration for the assessed removal procedure.
- [LOG] / [OBSERVATION] Evidence showing the accessible state before removal and the non-retrievable / non-usable state after removal through the interfaces and functions provided by the M&O system.

6.3.2.8.2 Assessment Case AC-M&O-DECOM-002

1. Assessment reference
Requirement: REQ-M&O-DECOM-002 (Basic - Single class)

2. Assessment objective
Verify that, where the M&O system supports transfer of orchestration data, management configuration, or associated metadata to another product or system, the M&O system protects the confidentiality and integrity of the transferred data through the interfaces and functions provided by the M&O system.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the supported transfer mechanism and the types of orchestration data, management configuration, or associated metadata that can be transferred.
- Documentation describing the security protections applied to the transfer.
- A test configuration in which supported data can be transferred from the M&O system to another product or system.
- Access to the interfaces and functions provided by the M&O system for the transfer operation.

4. Assessment activities
The assessment shall at least:

1) Review documentation to identify the supported transfer mechanism and the security protections applied to the transferred data.
2) Invoke the documented transfer mechanism for supported orchestration data, management configuration, or associated metadata.
3) Verify that the transfer is performed through the interfaces and functions provided by the M&O system with protection of the confidentiality and integrity of the transferred data.
4) Where feasible, verify that transfer without the documented protection is not permitted through the interfaces and functions provided by the M&O system.

5. Assessment verdict

- Pass: Where the M&O system supports transfer, the documented transfer mechanism protects the confidentiality and integrity of the transferred data through the interfaces and functions provided by the M&O system.
- Fail: A supported transfer mechanism does not protect the confidentiality or integrity of the transferred data; or transfer can be performed through the interfaces and functions provided by the M&O system without the documented protection.

6. Assessment evidence

- [DESIGN] Documentation of the transfer mechanism and associated security protections.
- [SCOPE] Identification of the transferred data types assessed.
- [CONFIG] Relevant M&O configuration for the assessed transfer procedure.
- [LOG] / [OBSERVATION] Evidence showing the protected transfer of the assessed data through the interfaces and functions provided by the M&O system.

6.4 CES

6.4.1 CRS

6.4.1.1 Assessment for Container Isolation

6.4.1.1.1 Assessment Case AC-CRS-CN-ISO-001

1. Assessment reference
Requirement: REQ-CRS-CN-ISO-001 (Basic)

2. Assessment objective
Verify that:

1) The CRS enforces isolation between containers using OS isolation mechanisms under its control, including at least process, filesystem, network and resource management layers.
2) The CRS supports configurations in which one container cannot read, write or otherwise interfere with another container's processes, filesystem contents, network endpoints or allocated resources.

3. Assessment preparation
The assessment shall have access to:

- Documentation of the CRS container isolation model (e.g. namespaces, cgroups/quotas, virtual networking, per-container filesystems).
- Documentation describing how containers are configured to achieve strict isolation (e.g. non-shared Process ID (PID) /Inter-Process Communication (IPC) /network / mount namespaces, independent root filesystems).
- A test environment with at least two containers running on the same CRS-managed host, under different identities/workloads, with the ability to execute commands inside each container.

4. Assessment activities
The assessment shall at least:

1) Process isolation:

- From container A, list processes and attempt to inspect or signal processes belonging to container B.
- Verify that only processes of container A are visible/controllable.

2) Filesystem isolation:

- From container A, enumerate filesystem paths and attempt to access directories or files that are part of container B's filesystem (other than intentionally shared volumes, if any).
- Verify that container A cannot read or modify container B's filesystem contents.

3) Network isolation:

- Identify network endpoints assigned to each container.
- From container A, attempt to bind to IP addresses/ports assigned to container B or otherwise interfere with B's network endpoints beyond normal, configured connectivity (e.g. service-level communication).
- Verify that container A cannot hijack or directly interfere with container B's endpoints.

4) Resource isolation:

- Review configured resource controls (e.g. CPU and memory limits) and generate load in container A.
- Verify that container B continues to operate as expected and that its resource allocations behave according to the configured limits, without arbitrary starvation caused by A beyond normal scheduling behaviour.

5. Assessment verdict

- Pass: documentation and tests confirm that process, filesystem, network and resource isolation are enforced, and that one container cannot read, write or otherwise interfere with another container's execution, data or configuration state in the assessed configuration.
- Fail: a container can see or control another container's processes, access its filesystem unexpectedly, interfere with its network endpoints or consume resources in a way that bypasses configured isolation.

6. Assessment evidence

- [SCOPE] Description of CRS deployment, host and containers used for the isolation assessment.
- [CONFIG] CRS configuration for namespaces, networking, storage and resource limits for the test containers.
- [DESIGN] Documentation of CRS container isolation mechanisms across process, filesystem, network and resource layers.
- [OBSERVATION] / [LOG] Command outputs/logs showing isolation behaviour and blocked cross-container access attempts.

6.4.1.1.2 Assessment Case AC-CRS-CN-ISO-002

1. Assessment reference
Requirement: REQ-CRS-CN-ISO-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS supports configuration of strong separation between containers and the host OS by restricting container access to kernel interfaces, privileged operations and system resources outside their assigned execution context.

2) The CRS can:

- limit container privileges (e.g. reduced kernel capabilities, prevention of access to host-level device nodes and sensitive filesystems); and
- restrict container access to kernel interfaces and system calls not required for the declared workload, using mechanisms such as syscall filtering, Mandatory Access Control (MAC) policies or equivalent controls.

3. Assessment preparation
The assessment shall have access to:

- Documentation of CRS mechanisms for strong container-host separation (e.g. capabilities model, device-node access controls, mount policies, syscall filtering, MAC policies).
- Documentation of typical secure configurations (profiles, templates or policies) for non-privileged containers.
- A test environment where containers can be created with differing privilege profiles (e.g. reduced capabilities, syscall filters, MAC labels).

4. Assessment activities
The assessment shall at least:

1) Privilege reduction

- Configure a container with a restricted privilege profile as recommended (e.g. reduced capabilities, non-privileged user, no direct device-node access).
- Inside the container, attempt operations that require elevated privileges or kernel capabilities (e.g. loading kernel modules, raw socket creation, mounting host filesystems, accessing /dev nodes that should be restricted).
- Verify that such operations are denied.

2) Host resource protection

- From the restricted container, attempt to access sensitive host filesystems or directories (e.g. host root, system configuration directories, host-level logs) that are not explicitly mounted into the container.
- Verify that these locations are not accessible.

3) Kernel interface and syscall restriction

- Configure syscall filtering or equivalent kernel-interface restriction (e.g. seccomp profiles, MAC rules) for the container according to documentation for the declared workload.
- From within the container, attempt to invoke known disallowed system calls or interact with kernel interfaces that the profile is intended to block.
- Verify that such calls are blocked (e.g. denied, terminated) while permitted system calls required for normal workload operation continue to function.

5. Assessment verdict

- Pass: the CRS can configure strong separation between containers and the host; privilege reduction mechanisms are effective; container access to device nodes and sensitive filesystems is restricted; and syscall/filtering/MAC-equivalent controls prevent access to non-required kernel interfaces while allowing necessary workload functions.
- Fail: containers can retain broad privileges, directly access sensitive host resources, or bypass syscall/interface restrictions in the assessed secure configuration.

6. Assessment evidence

- [CONFIG] Container profiles showing reduced capabilities, device/mount restrictions and syscall/MAC policies.
- [DESIGN] Documentation of CRS mechanisms for strong container-host separation and recommended secure profiles.
- [OBSERVATION] / [LOG] Outputs showing denied privileged operations, blocked access to host resources and enforcement of syscall/MAC restrictions.

6.4.1.1.3 Assessment Case AC-CRS-CN-ISO-003

1. Assessment reference
Requirement: REQ-CRS-CN-ISO-003 (Advanced)

2. Assessment objective
Verify that:

1) The CRS supports execution of containers within hardened isolation environments that strengthen separation between container workloads and the host kernel (e.g. micro-VM-based runtimes, user-space kernel sandboxes, additional Virtualisation/sandbox layers or equivalent).

2) In such hardened environments, compromise of a container does not directly grant an attacker equivalent access to the host kernel or to other containers.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing hardened isolation environments supported by the CRS for containers (e.g. architecture diagrams and isolation properties of micro-VM runtimes, sandbox layers).
- Documentation explaining configuration and operational conditions for enabling hardened isolation for selected workloads.
- A test configuration where at least one container can be deployed in a hardened isolation mode and another container in a standard mode on the same host.

4. Assessment activities
The assessment shall at least:

1) Hardened environment configuration:

- Configure a container to run using the CRS hardened isolation mechanism (e.g. micro-VM-backed container, user-space kernel sandbox).
- Verify from configuration/status that the container is indeed running with the hardened isolation mode enabled as documented.

2) Containment of compromise within hardened environment:

- Within the hardened-isolation container, attempt operations that would normally be indicative of host compromise if successful (e.g. direct interaction with host kernel interfaces, access to host filesystems, access to other containers' namespaces or storage, escalation to host-level privileges).
- Verify that such operations are confined or blocked by the hardened isolation layer and do not result in direct access to the host kernel or other containers.

3) Separation from other containers:

- With at least one additional container running (standard or hardened), attempt from the hardened container to interact with or control the other container beyond normal, configured communication (e.g. attach to its processes, access its filesystems, inject code).
- Verify that these attempts fail and that cross-container compromise via the hardened environment is prevented.

4) Administrative visibility:

- Verify that administrators can identify containers running with hardened isolation (e.g. via configuration, labels, status commands or logs), so that use of this mode is auditable and can be targeted to high-risk workloads.

5. Assessment verdict

- Pass: the CRS supports hardened isolation environments for containers; such environments can be configured for selected workloads; and in the assessed configuration, compromise of a container within a hardened environment does not directly provide equivalent access to the host kernel or other containers.
- Fail: hardened isolation mode is not available or cannot be configured; or a compromised container in a hardened environment can directly access the host kernel or other containers in ways inconsistent with the strengthened isolation objective.

6. Assessment evidence

- [CONFIG] CRS configuration showing containers deployed in hardened isolation mode and any associated policies.
- [DESIGN] Documentation of hardened isolation architecture, including threat model and separation properties with respect to the host kernel and other containers.
- [PLT-CAP] Where applicable, evidence of platform capabilities used to implement hardened environments (e.g. hardware or software Virtualisation features).
- [OBSERVATION] Logs/traces and test results demonstrating that host/kernel and cross-container access attempts from a hardened container are blocked or confined as designed.

6.4.1.2 Assessment for Control Plane Isolation

6.4.1.2.1 Assessment Case AC-CRS-CP-ISO-001

1. Assessment reference
Requirement: REQ-CRS-CP-ISO-001 (Basic)

2. Assessment objective
Verify that:

1) The CRS provides access-control mechanisms for its management and control interfaces.
2) The CRS can be configured such that containerized workloads cannot directly access or invoke CRS management or control functions.
3) Only authenticated and authorized entities can perform control or configuration actions.
4) CRS management sockets, control APIs and management daemons can be deployed so they are not reachable from within container namespaces.

3. Assessment preparation

The assessment shall have access to:

- Documentation of CRS management and control interfaces (local sockets, REST APIs, CLIs, management daemons) and their access-control mechanisms.
- Documentation of how these interfaces are bound/deployed (e.g. host network namespace, Unix domain sockets, host-only endpoints).
- A test environment with:
  - at least one administrative account or management client;
  - at least one container with its own namespaces;
  - the ability to run commands inside containers and on the host.

4. Assessment activities
The assessment shall at least:

1) Identify all CRS management/control interfaces and confirm that authentication and authorization controls are configured (e.g. accounts, roles, policies).

2) From an authorized management context (host or management workstation), perform control actions (e.g. create/delete container, change policy) and verify that successful access requires authentication and that authorization (role/permission) checks are enforced.

3) From within a container:

- enumerate reachable network endpoints and local sockets;

- attempt to connect to CRS management sockets, control APIs or management daemons using no credentials and container-level credentials;

- attempt to perform control/configuration actions.

Verify that:

- CRS management endpoints are not reachable from within container namespaces (e.g. bound only to host network namespace or host-only sockets); and

- container principals cannot perform management or control actions.

5. Assessment verdict

- Pass: access-control mechanisms exist; CRS can be deployed so that management/control interfaces are not reachable from container namespaces; and only authenticated, authorized entities can perform control/configuration actions.

- Fail: containers can reach and invoke CRS management/control functions; or management interfaces do not enforce proper authentication/authorization.

6. Assessment evidence

- [CONFIG] CRS management/control interface configuration, including bindings (host-only vs container-visible) and access-control settings.

- [DESIGN] Documentation of CRS management/control model and access-control mechanisms.

- [OBSERVATION] / [LOG] Evidence of successful authorized management actions and failed/blocked attempts from within containers to reach management endpoints or perform control actions.

6.4.1.2.2 Assessment Case AC-CRS-CP-ISO-002

1. Assessment reference
Requirement: REQ-CRS-CP-ISO-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS supports configuration of dedicated logical separation between CRS management and control traffic and container data-plane traffic.

2) The CRS supports the use of host-only communication endpoints, distinct network namespaces or dedicated management networks for its management/control interfaces, enabling isolation of CRS control traffic from container workloads.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing how CRS management/control interfaces can be exposed over host-only endpoints, distinct network namespaces or dedicated management networks.

- Documentation of recommended topologies for separating management/control traffic from container data-plane traffic.
- A test configuration with:
  - at least one management/control endpoint exposed via host-only or dedicated management connectivity;
  - at least one container data-plane network/namespace;
  - tools to inspect network namespaces, interfaces, routes and firewall rules.

4. Assessment activities
The assessment shall at least:

1) Configure the CRS so that management/control interfaces use one of the supported separation mechanisms (e.g. host-only socket, host network namespace only, dedicated management network or VLAN).
2) Verify via configuration and inspection that:
   - CRS management/control endpoints are bound only to the designated management/host-only mechanisms; and
   - they are not exposed on container data-plane networks or in container network namespaces.
3) From a container attached only to the data-plane network/namespace, attempt to connect to CRS management/control endpoints; verify that such attempts are blocked or cannot reach the endpoints.
4) From an administrative context on the host or on the dedicated management network, verify that management/control interfaces are reachable via the designated paths, and that container data traffic does not traverse those paths.

5. Assessment verdict

- Pass: CRS management/control traffic can be logically separated from container data-plane traffic using host-only endpoints, distinct network namespaces or dedicated management networks; and container data-plane networks cannot be used to reach CRS management/control interfaces in the assessed configuration.
- Fail: no effective mechanism to separate CRS management/control and container data-plane traffic; or management/control interfaces remain reachable over container data-plane networks despite intended separation.

6. Assessment evidence

- [CONFIG] Network and namespace configuration showing bindings of CRS management/control interfaces and separation from container data-plane networks.
- [DESIGN] Documentation of supported mechanisms for control-plane vs data-plane separation.
- [OBSERVATION] / [LOG] Connection tests from container and management contexts demonstrating blocked access from data-plane and permitted access via the management/host-only paths.

6.4.1.2.3 Assessment Case AC-CRS-CP-ISO-003

1. Assessment reference
Requirement: REQ-CRS-CP-ISO-003 (Advanced)

2. Assessment objective
Verify that:

1) The CRS supports cryptographically protected and mutually authenticated communication channels for all remote or network-exposed CRS management and control interfaces.
2) When such channels are used, management/control traffic is transmitted only over communication paths that provide confidentiality, integrity protection and mutual authentication between the CRS and authorized management entities.

3) For CRS management interfaces restricted to local host-only mechanisms, equivalent protection is provided by underlying platform access controls, consistent with the NOTE.

3. Assessment preparation

The assessment shall have access to:

- Documentation of remote/network-exposed CRS management/control interfaces and supported cryptographic protocols and mutual authentication options (client certificates, key-based methods).
- Documentation for configuring certificates/keys and enabling mutually authenticated channels.
- Documentation of host-only management mechanisms (e.g. Unix sockets, loopback-only endpoints) and the associated OS access-control model.
- A test configuration with:
  - at least one remote/network-exposed management/control interface;
  - representative host-only management mechanisms, if provided;
  - one management client with valid credentials and another with invalid/missing credentials.

4. Assessment activities

The assessment shall at least:

For remote/network-exposed management/control interfaces:

1) Configure each such interface to use a cryptographically protected transport with mutual authentication as documented.

2) Establish a management session using a client with valid credentials (certificates/keys) and verify via protocol inspection and logs that:

- the session is encrypted;
- the CRS authenticates the client; and
- the client authenticates the CRS (e.g. server certificate validation).

3) Attempt connections from clients without valid credentials or using disallowed protocol/cipher configurations and verify that these are rejected and cannot perform management/control actions.

4) Verify that no alternative cleartext or unauthenticated network paths exist for CRS management/control in the assessed configuration (e.g. no parallel HTTP port without TLS).

For host-only management mechanisms (where present):

1) Verify that such interfaces are bound only to host-local mechanisms (e.g. Unix domain sockets, loopback-only addresses) and not reachable from remote networks or container namespaces.

2) Verify that OS-level access controls (file permissions, user/group roles, MAC policies) restrict access to these host-only interfaces to authorized local principals, providing confidentiality, integrity and authenticity comparable to the cryptographic protections required on remote interfaces.

5. Assessment verdict

- Pass: all remote/network-exposed CRS management/control interfaces can use cryptographically protected, mutually authenticated channels; management/control traffic over these interfaces is confined to communication paths providing confidentiality, integrity and mutual authentication; and host-only management mechanisms rely on OS access controls that provide equivalent protection without exposing network-reachable cleartext admin paths.
- Fail: some remote management/control interfaces cannot be protected cryptographically or cannot use mutual authentication; cleartext or unauthenticated admin channels remain reachable; or host-only mechanisms are exposed or insufficiently protected.

6. Assessment evidence

- [CONFIG] Cryptographic and authentication configuration for remote management/control interfaces, and configuration for host-only management mechanisms (bindings, permissions).
- [DESIGN] Documentation of supported management/control protocols, mutual authentication mechanisms and security model for host-only interfaces.
- [LOG] / [OBSERVATION] Protocol traces and logs showing encrypted, mutually authenticated sessions, failed attempts without valid credentials, absence of cleartext admin paths, and enforcement of OS access controls on host-only interfaces.

6.4.1.3 Assessment for Network Plane Isolation

6.4.1.3.1 Assessment Case AC-CRS-NP-ISO-001

1. Assessment reference
Requirement: REQ-CRS-NP-ISO-001 (Basic)

2. Assessment objective
Verify that:

1) The CRS maintains logical separation between container network traffic and host/CRS management network traffic using virtual networking mechanisms under its control.
2) The CRS supports configurations in which containers cannot directly access host or CRS management interfaces/services by default.
3) Separate network contexts can be defined for management traffic and container workload traffic.

3. Assessment preparation

The assessment shall have access to:

- Documentation of the CRS networking model (container networks, host network, management endpoints, virtual switches, CNI/plug-ins, etc.).
- Documentation describing how management endpoints (APIs, daemons, sockets) are bound to host/management networks versus container networks.
- A test configuration with:
  - at least one container network with one or more containers;
  - CRS/host management interfaces bound to host/management network context;
  - tools to list network interfaces/namespaces and test connectivity from containers and host.

4. Assessment activities
The assessment shall at least:

1) Configure the CRS so that:
   - containers are attached to one or more container networks; and
   - CRS/host management interfaces are exposed only on host or dedicated management network contexts, as documented.
2) From within a container, enumerate network interfaces, routing tables and reachable IP/ports, and attempt to connect to:
   - CRS management APIs;
   - CRS management daemons;
   - host administrative services.

Verify that these interfaces/services are not reachable by default.

3) From a host/management context, verify that management interfaces are reachable via the intended management/host network and that container data-plane traffic uses separate network contexts.

5. Assessment verdict

- Pass: virtual networking under CRS control provides logical separation; containers cannot directly access host or CRS management interfaces/services by default; and separate network contexts for management and container workload traffic can be configured and observed.
- Fail: container traffic can reach host/CRS management interfaces by default; or management and container traffic share the same network context without separation options.

6. Assessment evidence

- [SCOPE] Description of the CRS deployment, container networks, and management/host networks used in the assessment.
- [CONFIG] CRS networking configuration (CNI or equivalent), interface bindings for management endpoints, container network definitions.
- [DESIGN] Documentation of CRS network plane separation model and default exposure of management vs container traffic.
- [OBSERVATION] / [LOG] Connectivity tests and command outputs from containers and host showing lack of default access from containers to management/host interfaces.

6.4.1.3.2 Assessment Case AC-CRS-NP-ISO-002

1. Assessment reference
Requirement: REQ-CRS-NP-ISO-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS supports configuration of explicit traffic filtering and access-control rules for:

- flows between containers;
- flows between container networks;
- flows between container networks and host/management networks.

2) These rules are enforced at the CRS virtual networking layer or other enforcement points under CRS control.

3) Only network flows explicitly permitted by policy are allowed to cross these boundaries.

3. Assessment preparation

The assessment shall have access to:

- Documentation of CRS mechanisms for defining and enforcing network policies (e.g. network policies, security groups, virtual firewall rules, CNI plug-in ACLs).
- Documentation with examples of permitted and denied flows.
- A test configuration with:
  - at least two container networks and multiple containers;
  - host/management network reachable from the CRS;
  - the ability to define and modify network policies and to generate traffic between these endpoints.

4. Assessment activities
The assessment shall at least:

1) Define and apply network policies or equivalent rules at the CRS virtual networking layer (or other CRS-controlled enforcement point) to:

- allow specific flows (e.g. container A in network X to container B in network Y on a given port; management endpoint to specific container agents); and

- deny all other cross-container, cross-network and container-to-host/management flows by default.

2) Generate test traffic:

- flows that match permitted rules;

- flows that should be blocked (e.g. arbitrary container-to-container connections across networks, container-to-management attempts outside permitted scope).

3) Verify that:

- permitted flows succeed as configured;

- non-permitted flows are blocked at the CRS-controlled enforcement points; and

- adjusting rules (add/remove) changes allowed/denied flows as expected.

5. Assessment verdict

- Pass: the CRS can define and enforce explicit traffic filtering/access-control rules between containers, container networks and host/management networks, and only flows explicitly permitted by policy can cross these boundaries.

- Fail: policies cannot be defined at the CRS-controlled layer; enforcement is ineffective; or cross-boundary traffic bypasses the configured rules.

6. Assessment evidence

- [CONFIG] Network policy and access-control rule definitions and their association with container networks and host/management networks.

- [DESIGN] Documentation of CRS virtual networking enforcement mechanisms for inter-container and cross-plane traffic.

- [OBSERVATION] / [LOG] Packet captures, policy logs or connectivity test results showing success of permitted flows and blocking of non-permitted flows.

6.4.1.3.3 Assessment Case AC-CRS-NP-ISO-003

1. Assessment reference
Requirement: REQ-CRS-NP-ISO-003 (Advanced)

2. Assessment objective
Verify that:

1) The CRS supports cryptographically protected and mutually authenticated communication channels for management and control traffic that traverses shared or untrusted networks.

2) When such channels are used, CRS management/control traffic is transmitted only over paths that provide confidentiality, integrity protection and mutual authentication between the CRS and authorized management entities.

3) For management interfaces restricted to local host-only mechanisms, equivalent protection is provided by OS platform access controls, consistent with the NOTE.

3. Assessment preparation
The assessment shall have access to:

- Documentation of remote/network-exposed CRS management/control interfaces and supported cryptographic protocols (e.g. TLS, mTLS, SSH, VPN/IPsec) and mutual authentication methods (client certificates, keys, tokens).
- Documentation explaining how to configure these protected channels and how they are bound to management/control traffic.
- Documentation of host-only management interfaces (e.g. Unix domain sockets, loopback-only endpoints) and their OS-level access-control model.
- A test configuration with:
  - at least one CRS management/control interface exposed over a shared or untrusted network;
  - at least one representative host-only management mechanism;
  - management clients with valid and invalid credentials.

4. Assessment activities
For management/control interfaces over shared or untrusted networks, the assessment shall at least:

1) Configure each such interface to use cryptographically protected transport with mutual authentication as documented (e.g. mTLS for control APIs, SSH with key-based admin access, VPN/IPsec tunnels for remote management).
2) From a client with valid credentials, establish a management/control session and verify via protocol inspection and logs that:
   - the session is encrypted;
   - the CRS authenticates the client;
   - the client authenticates the CRS.
3) Attempt connections from clients without valid credentials or using disallowed protocol or cipher configurations and verify that these are rejected and cannot perform management or control actions.
4) Verify that no cleartext or unauthenticated alternative management/control paths are reachable over the shared/untrusted network in the assessed configuration.

For host-only management interfaces, the assessment shall at least:

1) Verify that such interfaces are bound only to local mechanisms (e.g. Unix domain sockets, loopback-only addresses) and are not reachable from container namespaces or external networks.
2) Verify that OS-level access controls (file permissions, user/group roles, MAC policies) restrict access to these interfaces to authorized local principals, providing confidentiality, integrity and authenticity comparable to what cryptographic protection would provide on a network path.

5. Assessment verdict

- Pass: the CRS supports cryptographically protected, mutually authenticated channels for management/control traffic over shared/untrusted networks; such traffic is carried only over paths providing confidentiality, integrity and mutual authentication; and host-only management mechanisms rely on appropriate OS access controls without exposing unprotected network-reachable management paths.
- Fail: remote CRS management/control interfaces cannot be protected cryptographically or cannot use mutual authentication; cleartext or unauthenticated management/control paths exist over shared/untrusted networks; or host-only mechanisms are exposed beyond the host or lack sufficient OS-level protection.

6. Assessment evidence

- [CONFIG] Cryptographic and authentication configuration for remote CRS management/control interfaces; configuration for host-only interfaces (bindings, permissions).
- [DESIGN] Documentation of CRS control-plane security model, supported protocols, mutual authentication mechanisms and host-only protection model.
- [LOG] / [OBSERVATION] Protocol traces and logs demonstrating encrypted, mutually authenticated management sessions, rejected invalid attempts, absence of cleartext management traffic on shared networks, and enforcement of OS access controls for host-only mechanisms.

6.4.1.4 Assessment for Boot Integrity Verification

6.4.1.4.1 Assessment Case AC-CRS-B-INT-002

1. Assessment reference
Requirement: REQ-CRS-B-INT-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS supports deployment with a startup, loading, or installation mechanism that verifies the integrity and authenticity of its executable artefacts before those artefacts are used to enable container scheduling, container lifecycle operations, or container execution interfaces.
2) The CRS is delivered with, or supports the use of, verification metadata for those executable artefacts, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
3) The product security documentation identifies the startup, loading, or installation mechanism used or expected for the CRS executable artefacts, including any Operational Environment dependency and the expected security properties of that mechanism.
4) Where the CRS includes the mechanism that loads, activates, or starts its executable artefacts, this mechanism prevents loading, activation, or startup of those artefacts when verification fails.
5) Where verification status from a platform-provided startup, loading, or installation mechanism is made available to the CRS, the CRS uses that status before enabling container scheduling, container lifecycle operations, or container execution interfaces.
6) Where the required verification status cannot be established or is reported as failed, the CRS prevents enabling container scheduling, container lifecycle operations, or container execution interfaces where such enforcement is under the control of the CRS.

This assessment does not require the CRS to verify firmware, platform bootloaders, Host OS components, service managers, package managers, hardware roots of trust, or other Operational Environment components, unless those components are part of the declared CRS product.

3. Assessment preparation
The assessment shall have access to:

- Product security documentation identifying the CRS executable artefacts covered by REQ-CRS-B-INT-002.
- Documentation of the startup, loading, or installation mechanism used or expected for those artefacts.
- Verification metadata associated with the CRS executable artefacts, such as signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Documentation of any Operational Environment dependency used for verification, including expected security properties and assumptions required for the CRS to be securely started or loaded.

- Design, configuration, or interface evidence showing how the CRS uses verification status before enabling container scheduling, container lifecycle operations, or container execution interfaces, where such status is made available to the CRS.
- Evidence of failure behaviour where verification fails or required verification status cannot be established, where such enforcement is under the control of the CRS.

4. Assessment activities

The assessment shall at least:

1) Check that the product security documentation identifies the CRS executable artefacts covered by REQ-CRS-B-INT-002.
2) Check that the CRS is delivered with, or supports the use of, verification metadata for those executable artefacts.
3) Check that the product security documentation identifies the startup, loading, or installation mechanism used or expected for those executable artefacts.
4) Check whether the verification mechanism is provided by the CRS, by another component of the declared product, or by the Operational Environment.
5) Check that any Operational Environment dependency is identified together with its expected security properties and assumptions.
6) Where the CRS includes the mechanism that loads, activates, or starts its executable artefacts, check that this mechanism prevents loading, activation, or startup when verification fails.
7) Where verification status from a platform-provided startup, loading, or installation mechanism is made available to the CRS, check that the CRS uses this status before enabling container scheduling, container lifecycle operations, or container execution interfaces.
8) Where the required verification status cannot be established or is reported as failed, check that the CRS prevents enabling container scheduling, container lifecycle operations, or container execution interfaces where such enforcement is under the control of the CRS

5. Assessment verdict

- Pass: the required evidence is provided and demonstrates that the CRS supports deployment with a startup, loading, or installation mechanism that verifies the integrity and authenticity of its executable artefacts in accordance with REQ-CRS-B-INT-002, including documented Operational Environment dependencies and failure behaviour where enforcement is under the control of the CRS.
- Fail: the required evidence is missing or does not demonstrate that the CRS supports the required integrity and authenticity verification capability for its executable artefacts.

The verdict shall not be Fail solely because firmware, platform bootloaders, Host OS components, service managers, package managers, hardware roots of trust, or other Operational Environment components are not verified by the CRS, provided that such components are outside the declared CRS product and are documented as Operational Environment dependencies.

6. Assessment evidence

- [DESIGN]:
  - Documentation identifying the CRS executable artefacts covered by REQ-CRS-B-INT-002.
  - Documentation of the startup, loading, or installation mechanism used or expected.
  - Documentation of Operational Environment dependencies, expected security properties, and assumptions
- [CONFIG] Verification metadata associated with the covered CRS executable artefacts.
- [OBSERVATION] / [TEST]:
  - Evidence showing use of verification status where such status is made available to the CRS.

- Evidence showing failure behaviour where verification fails or required verification status cannot be established, where enforcement is under the control of the CRS.

6.4.1.5 Assessment for Container Image Integrity Verification

6.4.1.5.1 Assessment Case AC-CRS-IMG-INT-001

1. Assessment reference
Requirement: REQ-CRS-IMG-INT-001 (Basic)

2. Assessment objective
Verify that the CRS verifies the integrity of container images using trusted integrity metadata, such as declared cryptographic digests, trusted reference values, or equivalent integrity metadata, at image ingestion, before first use, or before container instantiation, according to the configured image trust policy.

Verify that where enforcement is under the control of the CRS, container instantiation from images whose integrity verification fails is prevented

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CRS image integrity verification mechanism.
- Documentation identifying the trusted integrity metadata used by the CRS, such as declared cryptographic digests, trusted reference values, manifest or layer digests, or equivalent integrity metadata.
- Documentation describing when image integrity verification is performed, such as at image ingestion, before first use, or before container instantiation, according to the configured image trust policy.
- Documentation identifying whether enforcement of failed image integrity verification is under the control of the CRS or depends on another component in the declared product or Operational Environment.
- A test environment with the CRS configured according to the documented image trust policy.
- Test images including:
  - one image with valid trusted integrity metadata;
  - one image with modified content or invalid integrity metadata;
  - one image without required integrity metadata, where applicable to the configured image trust policy.

4. Assessment activities
The assessment shall at least:

1) Review the CRS image integrity verification mechanism and the configured image trust policy.
2) Identify whether verification is performed at image ingestion, before first use, or before container instantiation.
3) Present a container image with valid trusted integrity metadata and verify that the image is accepted according to the configured image trust policy.
4) Present a container image with modified content or invalid trusted integrity metadata and verify that the CRS detects the integrity failure.
5) Where the configured image trust policy requires integrity metadata, present an image without the required integrity metadata and verify that the image is rejected or treated according to the documented policy.
6) Where enforcement is under CRS control, attempt to instantiate a container from an image whose integrity verification fails and verify that instantiation is prevented.
7) Where enforcement depends on another component in the declared product or Operational Environment, verify that the dependency and expected security properties are documented.

5. Assessment verdict

- Pass: the CRS verifies container image integrity using trusted integrity metadata, such as declared cryptographic digests, trusted reference values, or equivalent integrity metadata, at image ingestion, before first use, or before container instantiation according to the configured image trust policy; images with invalid integrity metadata are detected; and, where enforcement is under CRS control, container instantiation from images whose integrity verification fails is prevented. Where enforcement depends on another component in the declared product or Operational Environment, the dependency and expected security properties are documented.

- Fail: the CRS does not verify container image integrity using trusted integrity metadata according to the configured image trust policy; or images with invalid integrity metadata are not detected; or, where enforcement is under CRS control, containers can be instantiated from images whose integrity verification fails; or required dependencies for enforcement are not documented.

6. Assessment evidence

- [DESIGN] Documentation of the CRS image integrity verification mechanism, supported trusted integrity metadata, verification timing, and enforcement model.

- CONFIG] CRS image trust policy and integrity verification configuration.

- [LOG] / [OBSERVATION] Evidence that images with valid integrity metadata are accepted and images with invalid or missing required integrity metadata are detected or rejected according to policy.

- [LOG] / [OBSERVATION] Where enforcement is under CRS control, evidence that container instantiation is prevented when image integrity verification fails.

- [OE] Documentation of dependencies and expected security properties where enforcement depends on another component or Operational Environment mechanism.

6.4.1.5.2 Assessment Case AC-CRS-IMG-INT-002

1. Assessment reference
Requirement: REQ-CRS-IMG-INT-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS verifies both the integrity and the authenticity of container images before execution.

2) Authenticity verification establishes that each image originates from a trusted publisher or approval authority and has not been altered after approval.

3) The CRS prevents execution or instantiation of any container image whose integrity or authenticity cannot be successfully verified according to the configured trust policy.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the image trust model and verification mechanisms, including:
  - how trusted publishers or approval authorities are represented and configured;
  - how integrity and authenticity checks are combined (e.g. signatures, certificates, provenance metadata, equivalent cryptographic mechanisms).
- A test environment with:
  - at least one container image produced and "approved" according to the documented process (e.g. properly signed or otherwise bound to a trusted publisher);
  - one or more images for which integrity or authenticity will fail, such as:
    - an image modified after approval;

  - an image missing authenticity information;
  - an image signed or bound to an untrusted publisher, prepared in a controlled manner;
- a configured trust policy specifying which publishers/authorities are trusted.

4. Assessment activities
The assessment shall at least:

1) Review documentation and configuration to identify:
- the trust anchors or metadata used to represent trusted publishers/authorities;
- how the CRS enforces the trust policy when verifying images.
2) For an image that is correctly produced and approved under the trust policy:
- request creation or start of a container from that image;
- verify that both integrity and authenticity verification are applied and succeed;
- confirm that the container is instantiated.
3) For each test image that should fail integrity or authenticity (modified content, missing/invalid signature, untrusted publisher, or equivalent):
- request creation or start of a container from that image;
- verify that the CRS detects the failure to verify integrity and/or authenticity according to the configured trust policy;
- confirm that the CRS prevents execution or instantiation of the container from that image.

5. Assessment verdict

- Pass: integrity and authenticity verification is applied to container images before execution; only images from trusted publishers/authorities that have not been modified after approval can be instantiated; and images whose integrity or authenticity cannot be successfully verified under the trust policy are blocked.
- Fail: authenticity is not enforced or can be bypassed; untrusted or modified images can be instantiated; or failure to verify integrity/authenticity does not prevent container instantiation.

6. Assessment evidence

- [DESIGN] Documentation of the container image trust model, including integrity and authenticity verification and the trust policy.
- [CONFIG] Trust configuration (trusted publishers/authorities, key material, policy settings).
- [LOG] Records showing successful verification and container instantiation for approved images, and failed verification with blocked instantiation for images that do not satisfy the trust policy.
- [OBSERVATION] Test execution outputs demonstrating enforcement of the trust policy.

6.4.1.5.3 Assessment Case AC-CRS-IMG-INT-003

1. Assessment reference
Requirement: REQ-CRS-IMG-INT-003 (Advanced)

2. Assessment objective
Verify that:

1) The CRS performs integrity and authenticity verification of container images using trust anchors stored as protected trust material.
2) Protected trust material is not modifiable during normal operational state and cannot be modified by container workloads, unprivileged users, or routine runtime operations.

3) Protected trust material is updated or replaced only through a controlled, authenticated administrative process.

4) If verification of the integrity or authenticity of a container image using this protected trust material fails, or the verification status cannot be established, the CRS prevents execution or instantiation of that container image and records a security-relevant event.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing:
  - what constitutes protected trust material (keys, certificates, trust anchors);
  - how and where this material is stored;
  - what protections prevent modification during normal operation;
  - the process and required privileges for controlled update or replacement of trust material.
- A test environment with:
  - a CRS deployment configured to use protected trust material for image verification;
  - at least one container image that verifies successfully against this material;
  - the ability to simulate conditions where an image fails verification against the protected trust material or where verification status is indeterminate, in a controlled manner.

4. Assessment activities
The assessment shall at least:

1) Review documentation and configuration to confirm:
- which trust anchors are used for image verification;
- where they are stored and how access is restricted;
- that updates require a controlled, authenticated administrative process distinct from routine runtime operations.

2) In normal operational state, attempt to modify or replace trust material using:
- container workloads;
- unprivileged user contexts;
- standard administrative or runtime interfaces that are not part of the documented controlled update process.

3) Verify that such attempts fail or are blocked.

4) Using a container image correctly approved under the current protected trust material, request container creation/start and verify that:
- integrity and authenticity verification succeed;
- the container is instantiated.

5) Create a test condition such that verification against the protected trust material fails or becomes indeterminate for a container image (e.g. image modified after approval, image associated with no matching trust anchor) and then request container creation/start. Verify that:
- the CRS detects the verification failure or inability to establish verification status;
- the container image is not allowed to execute or be instantiated;
- a security-relevant event is recorded.

5. Assessment verdict

- Pass: the CRS uses protected trust material for image integrity/authenticity verification; this material cannot be modified by workloads, unprivileged users or routine operations; only controlled, authenticated processes can update it; and images that fail or cannot be verified against this material are blocked and generate a security-relevant event.
- Fail: trust anchors are stored in locations modifiable during normal operation; workloads or unprivileged users can alter them; or images that fail verification against the protected trust material can still be instantiated without a recorded security event.

6. Assessment evidence

- [DESIGN] Documentation of protected trust material, its storage, protection model and update procedures.
- [CONFIG] CRS configuration showing trust stores or equivalent structures used for image verification and their access-control settings.
- [LOG] Records showing successful verification and container instantiation for valid images, and failed verification with corresponding security-relevant events for images that cannot be verified.
- [OBSERVATION] Test outputs demonstrating blocked modification attempts on trust material and blocked instantiation of images that fail verification.

6.4.1.6 Assessment for Runtime Integrity Protection

6.4.1.6.1 Assessment Case AC-CRS-RP-INT-002

1. Assessment reference
Requirement: REQ-CRS-RP-INT-002 (Elevated)

2. Assessment objective
Verify that:

1) the CRS supports integrity protection for its security-critical runtime code, configuration state, and control logic used to enforce security policies or isolation;
2) the CRS supports mechanisms ensuring that security-critical configuration state and control logic can be modified only through authenticated and authorized interfaces or processes;
3) where runtime integrity protection relies on mechanisms provided by other components of the declared product or by the Operational Environment, the dependency is identified in the product security documentation.

3. Assessment preparation
The assessment shall have access to:

- product security documentation identifying the security-critical runtime code, configuration state, and control logic covered by REQ-CRS-RP-INT-002;
- design or configuration documentation describing the mechanisms used to protect the integrity of the covered runtime code, configuration state, and control logic;
- documentation identifying the interfaces or processes through which security-critical configuration state and control logic can be modified;
- documentation showing that such modification interfaces or processes require authentication and authorization;
- documentation of any dependency on other components of the declared product or on the Operational Environment, including expected security properties.

4. Assessment activities
The assessment shall at least:

1) Check that the product security documentation identifies the security-critical runtime code, configuration state, and control logic covered by REQ-CRS-RP-INT-002.

2) Check that the CRS supports mechanisms to protect the integrity of the identified security-critical runtime code, configuration state, and control logic.

3) Check that security-critical configuration state and control logic can be modified only through identified interfaces or processes.

4) Check that the identified modification interfaces or processes require authentication.

5) Check that the identified modification interfaces or processes require authorization.

6) Where runtime integrity protection relies on other components of the declared product or on the Operational Environment, check that the dependency and expected security properties are identified in the product security documentation.

5. Assessment verdict

- Pass: the required evidence demonstrates that the CRS supports integrity protection for its security-critical runtime code, configuration state, and control logic, and that security-critical configuration state and control logic can be modified only through authenticated and authorized interfaces or processes.
- Fail: the required evidence is missing or does not demonstrate that the CRS supports the required runtime integrity protection mechanisms or authenticated and authorized modification paths.

6. Assessment evidence

- [DESIGN]:
  - Documentation identifying the covered security-critical runtime code, configuration state, and control logic.
  - Description of the runtime integrity protection mechanisms.
  - Operational Environment dependency documentation, where applicable.
- [CONFIG] Documentation of authenticated and authorized modification interfaces or processes.

6.4.1.7 Assessment for Remote Attestation

6.4.1.7.1 Assessment Case AC-CRS-RA-INT-003

1. Assessment reference
Requirement: REQ-CRS-RA-INT-003 (Advanced, optional capability)

2. Assessment objective
Verify that, where remote attestation is included in the declared product configuration:

1) the CRS provides generation of attestation evidence describing launch-time measurements of its security-critical executable artefacts;

2) the attestation evidence is cryptographically verifiable by an authorized verifier;

3) the attestation evidence is bound to the CRS instance for which the measurements were produced;

4) the CRS protects attestation keys and measurement data against unauthorized access, disclosure, or replay;

5) the attestation evidence is limited to the minimum information necessary to verify the integrity of the measured CRS executable artefacts;

6) where attestation evidence relies on platform, Host OS, firmware, hardware root of trust, or external attestation services outside the declared product boundary or outside the responsibility of the manufacturer, those mechanisms are identified as Operational Environment dependencies in the product security documentation.

This assessment does not require attestation of CRS runtime configuration state, runtime policy state, arbitrary CRS memory, container memory, container workload state, transient execution state, workload-dependent runtime state, or other dynamic runtime data.

3. Assessment preparation
The assessment shall have access to:

- Product security documentation identifying whether remote attestation is included in the declared product configuration;
- Documentation identifying the CRS executable artefacts covered by attestation evidence;
- Documentation of the launch-time measurements generated or exposed by the CRS;
- Representative attestation evidence or attestation evidence format;
- Documentation showing how the attestation evidence is cryptographically verifiable by an authorized verifier;
- Documentation showing how the attestation evidence is bound to the CRS instance for which the measurements were produced;
- Documentation of mechanisms protecting attestation keys and measurement data against unauthorized access, disclosure, or replay;
- Documentation identifying any Operational Environment dependency used for attestation, including expected security properties.

4. Assessment activities
The assessment shall at least:

1) Check whether remote attestation is included in the declared product configuration.
2) If remote attestation is not included in the declared product configuration, assign Not Applicable for this assessment case.
3) Check that the attestation evidence describes launch-time measurements of security-critical CRS executable artefacts.
4) Check that the attestation evidence is cryptographically verifiable by an authorized verifier.
5) Check that the attestation evidence is bound to the CRS instance for which the measurements were produced.
6) Check that attestation keys and measurement data are protected against unauthorized access, disclosure, or replay.
7) Check that the attestation evidence is limited to the minimum information necessary to verify the integrity of the measured CRS executable artefacts.
8) Check that any platform, Host OS, firmware, hardware root of trust, or external attestation service relied upon for attestation is identified as an Operational Environment dependency, including expected security properties.
9) Check that the assessment does not require attestation of CRS runtime configuration state, runtime policy state, arbitrary CRS memory, container memory, container workload state, transient execution state, workload-dependent runtime state, or other dynamic runtime data.

5. Assessment verdict

- Pass: remote attestation is included in the declared product configuration and the required evidence demonstrates that the CRS generates cryptographically verifiable attestation evidence for launch-time measurements of security-critical executable artefacts in accordance with REQ-CRS-RA-INT-003.
- Fail: remote attestation is included in the declared product configuration and the required evidence is missing or does not demonstrate compliance with REQ-CRS-RA-INT-003.
- Not Applicable: remote attestation is not included in the declared product configuration.

6. Assessment evidence

- [DESIGN]:
  - Documentation identifying the CRS executable artefacts covered by attestation evidence.
  - Description of launch-time measurements.
  - Cryptographic verifiability and instance-binding description.
  - Attestation key and measurement data protection evidence.
  - Operational Environment dependency documentation, where applicable.
- [CONFIG] Representative attestation evidence or attestation evidence format.

6.4.1.7.2 Assessment Case AC-CRS-RA-INT-004

1. Assessment reference
Requirement: REQ-CRS-RA-INT-004 (Advanced, optional capability)

2. Assessment objective
Verify that, where CRS attestation evidence is intended to be used in remote verification scenarios and this capability is claimed:

1) The CRS supports configuration of privacy-preserving attestation options that reduce linkability and correlation between attestation events.
2) These privacy-preserving options preserve the ability of authorized verifiers to assess the integrity and configuration state of the CRS.
3) The CRS allows configuration of attested information, triggering conditions, and attestation intervals in order to support data minimisation and to prevent excessive or unauthorized monitoring.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing:
  - the CRS remote attestation model, including how CRS identity and instance information are represented in attestation evidence;
  - the privacy-preserving options available (for example limiting attested information, configuration of attestation triggers and intervals, use of pseudonymous identifiers or separation of identity and integrity information);
  - how these options are configured;
  - the expected impact of each option on attestation evidence fields, attestation frequency, and verifier behaviour.
- A test configuration with:
  - a CRS deployment where remote attestation to one or more authorized verifiers is enabled;
  - the ability to configure and modify the documented privacy-preserving options;

- tools or procedures to trigger multiple attestation events (for example on-demand, periodic, or event-driven) and capture the resulting attestation evidence under different privacy and scheduling configurations;

- access to an authorized verifier or verification tool capable of validating integrity and configuration state from the attestation evidence in both baseline and privacy-enhanced modes.

4. Assessment activities

The assessment shall at least:

1) Review documentation and CRS settings to identify:

- the default behaviour of remote attestation when privacy-preserving options are not enabled;

- the available privacy-preserving options and their documented effects on identifiers, metadata, attested content, and attestation frequency or trigger conditions;

- any prerequisites or constraints for enabling and configuring these options.

2) With privacy-preserving options disabled or in their default state:

- trigger multiple attestation events over time according to the default model;

- collect the corresponding attestation evidence;

- analyse the evidence to determine the stability and linkability of identifiers and other potentially identifying fields across events, and the amount of configuration and runtime information included;

- use the authorized verifier or tool to confirm that the integrity and configuration state of the CRS can be correctly assessed from this baseline evidence set.

3) Enable and configure the documented privacy-preserving options (for example limiting data fields, adjusting attestation triggers or intervals, enabling pseudonymous identifiers or separation of identity and integrity information) as described by the manufacturer, then:

- trigger multiple attestation events under the privacy-enhanced configuration;

- collect the resulting attestation evidence;

- compare this evidence set with the baseline to verify that linkability and correlation between attestation events are reduced in the documented manner and that attested content is limited in line with data-minimisation objectives;

- verify, using the authorized verifier or tool, that the integrity and configuration state of the CRS can still be reliably assessed using the privacy-enhanced evidence and any associated metadata or policies.

5. Assessment verdict

- Pass: when remote attestation is used and privacy support is claimed, the CRS provides configurable privacy-preserving attestation options that limit attested information, allow control of attestation triggers and intervals, and, where applicable, support techniques such as pseudonymous identifiers or separation of identity and integrity information; when enabled, these options reduce linkability and correlation between attestation events as documented while still allowing authorized verifiers to correctly assess the integrity and configuration state of the CRS.

- Fail: privacy-preserving options are not available despite being claimed; configuration changes do not meaningfully reduce linkability or correlation between attestation events or do not support data minimisation and control of attestation frequency; or enabling these options prevents authorized verifiers from correctly assessing the integrity and configuration state of the CRS.

6. Assessment evidence

- [DESIGN] Documentation of the CRS remote attestation model, identity representation, and privacy-preserving options, including their expected effects on attestation evidence, frequency, and verifier behaviour.

- [CONFIG] CRS and verifier configuration showing remote attestation settings, selected privacy-preserving options, and configured triggers or intervals.
- [LOG] Records of attestation events and related configuration changes, including timestamps and any metadata relevant to privacy options and scheduling.
- [OBSERVATION] Sets of attestation evidence collected before and after enabling privacy-preserving options, with verifier outputs showing successful integrity and configuration assessment in both modes and reduced linkability or correlation when privacy options are enabled.

6.4.1.8 Assessment for Administrative Authentication

6.4.1.8.1 Assessment Case AC-CRS-ADMIN-AUTH-001

1. Assessment reference
Requirement: REQ-CRS-ADMIN-AUTH-001 (Basic)

2. Assessment objective
Verify that:

1) CRS administrative interfaces require authentication before any administrative operation.
2) Administrative accounts use unique, non-default credentials and default credentials are disabled or changed before use.
3) Password complexity and lockout mechanisms for administrative accounts are configurable and enforced to mitigate brute-force attempts.
4) Where password-based authentication is used, administrative passwords are stored only as salted hashes.

3. Assessment preparation
The assessment shall have access to:

- Declared CRS product composition and list of administrative interfaces (CLI, APIs, dashboards/consoles).
- Documentation on administrative account model, default credentials, password policy and lockout options, and password storage mechanism.
- At least one CRS configuration with administrative accounts and access to identified administrative interfaces.

4. Assessment activities
The assessment shall at least:

1) Identify administrative interfaces from documentation/configuration and verify that each requires authentication in the assessed configuration.
2) Review installation/hardening guidance and verify in the assessed configuration that any default administrative credentials are disabled or enforced to be changed before administrative access is granted.
3) Confirm that password complexity and lockout policies for administrative accounts can be configured and verify that non-compliant passwords are rejected and that lockout is applied according to the configured thresholds.
4) Review documentation and, where feasible, configuration/implementation details to confirm that CRS administrative passwords are stored only as salted hashes (no cleartext).

5. Assessment verdict

- Pass: CRS administrative interfaces require authentication; administrative accounts do not rely on unchanged default credentials; password complexity and lockout are configurable and enforced; and administrative passwords are stored only as salted hashes.
- Fail: any administrative interface allows unauthenticated administrative operations; or unchanged default credentials can be used; or password complexity/lockout cannot be configured or are not enforced; or administrative passwords are stored in cleartext.

6. Assessment evidence

- [SCOPE] CRS product composition and list of administrative interfaces.
- [CONFIG] Administrative authentication, password policy and lockout configuration.
- [DESIGN] Description of administrative access model, default account handling and password storage mechanism.
- [LOG] / [OBSERVATION] Evidence of authentication enforcement, rejection of weak passwords and lockout behaviour.

6.4.1.8.2 Assessment Case AC-CRS-ADMIN-AUTH-002

1. Assessment reference
Requirement: REQ-CRS-ADMIN-AUTH-002 (Elevated)

2. Assessment objective
Verify that:

1) CRS administrative interfaces require authenticated access and support cryptographic key-based authentication for administrative accounts.
2) Password-based login for CRS administrative interfaces can be disabled.
3) Private keys used for CRS administrative authentication are protected from extraction or reuse by container workloads or other unprivileged components.
4) Algorithms and key lengths for key-based administrative authentication conform to the cryptographic profile declared by the manufacturer, aligned with Annex K.

3. Assessment preparation
The assessment shall have access to:

- Documentation on key-based administrative authentication for each CRS administrative interface.
- Documentation on applicable cryptographic profiles for administrative authentication (algorithms, key lengths, disallowed options).
- Documentation of key management and key storage for administrative private keys, including separation from container workloads and unprivileged components.
- At least one CRS configuration where administrative accounts use key-based authentication on relevant interfaces and password-based login can be disabled.

4. Assessment activities
The assessment shall at least:

1) Confirm from documentation/configuration that administrative accounts can use key-based authentication on relevant CRS interfaces and verify that an administrative account can obtain administrative access using the key-based mechanism.
2) Confirm that the CRS can disable password-based login for administrative interfaces and verify that, where disabled, password-only attempts are rejected while key-based authentication still permits authorized access.
3) Review configuration and, where applicable, protocol characteristics to confirm that algorithms, key lengths and protocol versions used for key-based administrative authentication comply with the declared cryptographic profile and that weaker parameters marked as non-compliant are not used.
4) Review key storage design and configuration to confirm that container workloads and other unprivileged components cannot read, export or misuse administrative private keys.

5. Assessment verdict

- Pass: CRS administrative interfaces require authenticated access and support key-based administrative authentication; password-based login can be disabled and is effectively blocked where configured; cryptographic parameters comply with the declared profile; and administrative private keys are protected from container workloads and unprivileged components.
- Fail: key-based administrative authentication is not available or not functional; or password-based login cannot be disabled or remains effective when disabled; or cryptographic parameters are weaker than the declared profile; or administrative private keys are accessible to container workloads or other unprivileged components.

6. Assessment evidence

- [CONFIG] CRS configuration for key-based administrative authentication, password-login disablement and crypto settings.
- [DESIGN] Documentation of key management and key storage architecture and applied cryptographic profile.
- [LOG] / [OBSERVATION] Evidence of successful key-based administrative authentication and rejection of password-only attempts when disabled.
- [PLT-CAP] Where relevant, evidence of platform capabilities used for key protection.

6.4.1.8.3 Assessment Case AC-CRS-ADMIN-AUTH-003

1. Assessment reference
Requirement: REQ-CRS-ADMIN-AUTH-003 (Advanced)

2. Assessment objective
Verify that:

1) the CRS requires strong authentication for all administrative interfaces;
2) where certificates are used for administrative authentication, the CRS verifies certificate validity, expiration, and revocation status before administrative access is granted.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the strong authentication mechanisms supported for CRS administrative interfaces and how they are enforced.
- Where certificates are supported for administrative authentication, documentation describing trusted CAs, certificate validation, expiration handling, and revocation checking mechanisms.
- A CRS configuration with administrative interfaces enabled and at least one administrative account or credential set using the supported strong authentication mechanism.
- Where certificates are used, valid and invalid, expired, and, where supported, revoked certificates or equivalent test conditions

4. Assessment activities
The assessment shall at least:

1) Identify the CRS administrative interfaces in scope and verify that each requires the configured strong authentication mechanism before administrative access is granted.
2) Attempt administrative access with valid strong-authentication credentials and verify that access is granted only after successful authentication.
3) Attempt administrative access with missing, invalid, or otherwise non-compliant authentication factors and verify that access is denied.
4) Where certificates are used for administrative authentication, verify that certificates are checked against the configured trust anchors and that invalid, expired, and, where supported, revoked certificates are rejected before access is granted.

5. Assessment verdict

- Pass: all administrative interfaces require strong authentication; and, where certificates are used for administrative authentication, certificate validity, expiration, and revocation status are verified before administrative access is granted.
- Fail: one or more administrative interfaces do not require strong authentication; or, where certificates are used, validity, expiration, or revocation status is not verified before administrative access is granted.

6. Assessment evidence

- [CONFIG] CRS configuration for strong administrative authentication, and, where applicable, certificate validation, trust-anchor, and revocation settings.
- [DESIGN] Documentation of the CRS administrative authentication architecture.
- [LOG] / [OBSERVATION] Evidence of successful access using valid strong-authentication credentials and rejection of invalid authentication attempts, including invalid, expired, or revoked certificates where certificates are used.

6.4.1.9 Assessment for Service Authentication

6.4.1.9.1 Assessment Case AC-CRS-SERV-AUTH-001

1. Assessment reference
Requirement: REQ-CRS-SERV-AUTH-001 (Basic)

2. Assessment objective
Verify that:

1) External services accessing CRS APIs or control interfaces (e.g. orchestration, monitoring, backup) are authenticated.
2) Each service account uses distinct, non-default credentials and no built-in shared or hard-coded service credentials are relied upon.
3) Configurable lockout and/or rate-limit mechanisms exist for service authentication attempts and are enforced.
4) Service credentials are stored in a way that prevents exposure via configuration files, container images or logs.

3. Assessment preparation
The assessment shall have access to:

- Declared CRS product composition and list of external services/interfaces accessing CRS APIs or control interfaces.
- Documentation on the service account model, credential provisioning, any default/built-in service credentials, and authentication lockout/rate-limiting options.
- Documentation on how service credentials are stored (e.g. secrets store, environment variables, KMS) and how configuration, images and logs are handled.
- At least one configuration with representative external services integrated using service accounts.

4. Assessment activities
The assessment shall at least:

1) Identify external services accessing CRS APIs or control interfaces and verify that each authenticates successfully before invoking management or control operations.
2) Review configuration and documentation to confirm that each service account has distinct credentials and that normal operation does not require shared or hard-coded service credentials; verify for at least one service that its credentials are unique and changeable.

3) Confirm that lockout and/or rate-limit thresholds for service authentication attempts can be configured; verify that repeated failed authentication attempts from a service trigger lockout or rate-limiting as configured.

4) Review the documented storage locations and mechanisms for service credentials and inspect representative configuration files, container images and logs to confirm that credentials are not stored or exposed in cleartext or other directly reusable form in these artefacts.

5. Assessment verdict

- Pass: external services accessing CRS APIs or control interfaces require authentication; each service account uses distinct, non-default credentials with no reliance on shared/hard-coded credentials; lockout or rate-limit mechanisms for service authentication attempts are configurable and enforced; and service credentials are not exposed via configuration files, container images or logs.
- Fail: unauthenticated service access to CRS APIs/control interfaces is possible; or shared/default/hard-coded service credentials are required or used; or lockout/rate-limit mechanisms are not available, not configurable, or not enforced; or service credentials are exposed in configuration files, images or logs.

6. Assessment evidence

- [SCOPE] List of external services and CRS APIs/control interfaces in scope.
- [CONFIG] Service authentication, credential, lockout and rate-limiting configuration.
- [DESIGN] Documentation of the service account and credential storage model, including handling of configuration, images and logs.
- [LOG] / [OBSERVATION] Evidence of enforced authentication, triggered lockout or rate-limiting, and absence of exposed credentials in sampled configuration files, images and logs.

6.4.1.9.2 Assessment Case AC-CRS-SERV-AUTH-002

1. Assessment reference
Requirement: REQ-CRS-SERV-AUTH-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS supports mutual cryptographic authentication for service-to-CRS interactions that access management functions or data, allowing both parties to verify each other's identity.

2) Cryptographic materials used for service authentication (keys, certificates, tokens) are protected from unauthorized extraction or reuse by container workloads or other unprivileged components.

3) Algorithms and key lengths used for service authentication conform to the cryptographic profile declared by the manufacturer, aligned with Annex K.

3. Assessment preparation
The assessment shall have access to:

- Documentation on mutual cryptographic authentication mechanisms for service-to-CRS interactions (e.g. mTLS, signed tokens).
- Documentation on applicable cryptographic profiles for service authentication (approved algorithms, minimum key lengths, disallowed options).
- Documentation on key/token management and storage (e.g. keystores, HSM/KMS, secret stores) and how separation from container workloads/unprivileged components is enforced.
- At least one configuration where representative services access CRS management functions or data using mutual cryptographic authentication.

4. Assessment activities
The assessment shall at least:

1) Confirm from documentation/configuration that mutual cryptographic authentication is supported for service-to-CRS interactions that access management functions or data; verify in the assessed configuration that a service using mutual authentication can establish an authenticated session and access management functions or data only after successful mutual verification.

2) Review configuration/protocol traces to confirm that algorithms, key lengths and protocol versions used for mutual authentication comply with the declared cryptographic profile and that non-compliant algorithms or key sizes are not negotiated.

3) Review the design and configuration of storage for service authentication keys/certificates/tokens and confirm that container workloads and other unprivileged components cannot read, export or misuse these materials according to documented isolation and access-control mechanisms.

5. Assessment verdict

- Pass: mutual cryptographic authentication is available and functional for service-to-CRS interactions that access management functions or data; cryptographic parameters used conform to the declared profile; and keys/certificates/tokens used for service authentication are protected from extraction or reuse by container workloads and other unprivileged components.

- Fail: mutual cryptographic authentication is not available or not functional where required; or cryptographic parameters are weaker than the declared profile; or cryptographic materials for service authentication can be accessed or reused by container workloads or other unprivileged components.

6. Assessment evidence

- [CONFIG] Mutual authentication configuration for services (certificates/keys/tokens and crypto settings).

- [DESIGN] Documentation of service authentication architecture, key/token management and storage, and applied cryptographic profile.

- [LOG] / [OBSERVATION] Evidence of successful mutual authentication and rejection of connections using non-compliant parameters or missing authentication.

- [PLT-CAP] Where relevant, evidence of platform capabilities used to protect service authentication materials.

6.4.1.9.3 Assessment Case AC-CRS-SERV-AUTH-003

1. Assessment reference
Requirement: REQ-CRS-SERV-AUTH-003 (Advanced)

2. Assessment objective
Verify that:

1) The CRS enforces a verifiable binding between service identities and cryptographic credentials for service interactions that access management functions or data.

2) Service credentials are issued, rotated and revoked through trusted credential or identity management systems.

3) The CRS automatically rejects revoked or expired service credentials during authentication.

4) The CRS supports integration with external identity or credential management systems to provide traceability of service identities across deployments and to ensure that only services with valid, managed identities can invoke CRS management or control functions.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing how service identities are represented (e.g. certificate subject/SAN, token claims, identity IDs) and how they are bound to cryptographic credentials.

- Documentation of trusted credential/identity management systems used (PKI, IdP, secret/credential manager) and the processes for issuing, rotating and revoking service credentials.

- Documentation on how the CRS validates credential status (expiry, revocation, or equivalent) during authentication.
- Documentation on integration with external identity/credential management for traceability (e.g. logs, mappings, cross-deployment identifiers).
- At least one configuration where service credentials are managed by an external trusted system, and valid, expired and revoked credentials (or equivalent conditions) can be exercised.

4. Assessment activities
The assessment shall at least:

1) Confirm that the CRS binds service identities to cryptographic credentials (e.g. through certificate identity or token claims) and verify that, in the assessed configuration, access to management or control functions depends on presenting a credential that matches the bound identity.
2) Verify that service credentials used with the CRS are issued, rotated and revoked by a trusted credential/identity management system; confirm, for at least one service, that credential updates (rotation) and revocation performed in the external system are reflected in CRS behaviour.
3) Verify that the CRS checks credential expiry and revocation (or equivalent status) during authentication and automatically rejects expired or revoked service credentials before granting access.
4) Verify that integration with the external identity/credential management system allows tracing service identities across deployments (e.g. consistent identifiers or mappings) and that attempts to invoke CRS management or control functions from services without valid, managed identities are denied.

5. Assessment verdict

- Pass: the CRS enforces verifiable binding of service identities to cryptographic credentials for service interactions accessing management functions or data; service credentials are managed (issued, rotated, revoked) by trusted systems; revoked or expired credentials are automatically rejected; and integration with external identity/credential management provides traceability and ensures that only services with valid, managed identities can invoke CRS management or control functions.
- Fail: service identity is not reliably bound to cryptographic credentials; or service credentials are not managed by trusted external systems; or revoked/expired credentials are not automatically rejected; or integration with external identity/credential management does not provide effective traceability or allow the CRS to block services without valid, managed identities.

6. Assessment evidence

- [CONFIG] CRS configuration for service identity binding, use of externally managed credentials and authentication status checks.
- [DESIGN] Documentation of the service identity and credential-lifecycle model, including integration with external identity/credential management and cross-deployment identity handling.
- [LOG] / [OBSERVATION] Evidence of successful authentication with valid managed credentials, rejection of expired/revoked credentials, and denial of access for services without valid managed identities.
- [PLT-CAP] Where relevant, evidence of platform capabilities used to support identity-bound credentials and status checking (e.g. PKI libraries, identity client libraries).

6.4.1.10 Assessment for Administrative Authorization

6.4.1.10.1 Assessment Case AC-CRS-ADMIN-AUTHZ-001

1. Assessment reference
Requirement: REQ-CRS-ADMIN-AUTHZ-001 (Basic)

2. Assessment objective
Verify that:

1) All administrative actions are restricted to authenticated administrative accounts.

2) A default-deny policy is enforced for access to CRS management functions, permitting only explicitly authorized actions.

3. Assessment preparation

The assessment shall have access to:

- Documentation of CRS administrative accounts, roles and management interfaces.
- Documentation of the administrative authorization model, including default behaviour.
- At least one configuration with multiple administrative accounts and access to CRS management functions.

4. Assessment activities
The assessment shall at least:

1) Verify that unauthenticated users and non-administrative accounts cannot perform CRS administrative actions on any management interface.

2) Confirm from configuration and tests that administrative actions not explicitly granted to an authenticated administrative account are denied by default.

5. Assessment verdict

- Pass: only authenticated administrative accounts can perform CRS administrative actions and any non-authorized administrative action is denied by default.
- Fail: unauthenticated/non-administrative users can perform administrative actions; or actions not explicitly authorized are allowed.

6. Assessment evidence

- [CONFIG] Administrative authorization and default policy configuration.
- [DESIGN] Description of CRS administrative authorization model and default-deny behaviour.
- [LOG] / [OBSERVATION] Evidence of denied operations for unauthenticated/non-admin users and permitted operations only when explicitly authorized.

6.4.1.10.2 Assessment Case AC-CRS-ADMIN-AUTHZ-002

1. Assessment reference
Requirement: REQ-CRS-ADMIN-AUTHZ-002 (Elevated)

2. Assessment objective
Verify that the CRS enforces granular authorization for administrative users and administrative operations, binding permitted actions to authenticated identities and enforcing least-privilege permissions by default. The assessment shall not require a specific authorization model, provided that the implemented model supports granular permissions and least-privilege enforcement. The implemented authorization model may be role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the implemented CRS administrative authorization model, including whether it is role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.
- Documentation identifying administrative operations protected by the authorization model.
- Documentation identifying authenticated administrative identities, permissions, policy rules, or equivalent authorization bindings.

- A test configuration with at least two authenticated administrative identities or accounts having different permissions for representative CRS administrative operations.
- Tools or interfaces to attempt permitted and non-permitted CRS administrative operations using those authenticated identities.

4. Assessment activities
The assessment shall at least:

1) Identify the CRS administrative operations covered by REQ-CRS-ADMIN-AUTHZ-002.
2) Review the implemented authorization model and confirm that permitted administrative actions are bound to authenticated identities.
3) Verify that least-privilege permissions are enforced by default for administrative identities or accounts.
4) Attempt a permitted CRS administrative action using an authenticated identity authorized for that action and verify that the action succeeds.
5) Attempt the same CRS administrative action using an authenticated identity not authorized for that action and verify that the action is denied.
6) Verify that the evidence demonstrates granular authorization without requiring a specific authorization model.

5. Assessment verdict

- Pass: the CRS enforces granular authorization for administrative operations, binds permitted administrative actions to authenticated identities, and enforces least-privilege permissions by default, regardless of whether the implemented model is role-based, attribute-based, policy-based, capability-based, or equivalent.
- Fail: the CRS does not enforce granular authorization for administrative operations; or permitted administrative actions are not bound to authenticated identities; or least-privilege permissions are not enforced by default; or an authenticated identity can perform an administrative action for which it is not authorized.

6. Assessment evidence

- [DESIGN] Documentation of the implemented CRS administrative authorization model and its binding of permissions to authenticated identities.
- [CONFIG] Authorization policy, role, attribute, rule, capability, or equivalent configuration showing permitted and non-permitted administrative actions.
- [LOG] / [OBSERVATION] Evidence of permitted administrative actions succeeding for authorized identities and non-permitted administrative actions being denied for unauthorized identities.

6.4.1.10.3 Assessment Case AC-CRS-ADMIN-AUTHZ-003

1. Assessment reference
Requirement: REQ-CRS-ADMIN-AUTHZ-003 (Advanced)

2. Assessment objective
Verify that the CRS enforces fine-grained, policy-driven, or context-aware authorization for administrative operations, including separation of duties, contextual policy evaluation, or integration with external identity, access-management, or policy decision systems where applicable to the corresponding requirement and where such integration is part of the declared product configuration. The assessment shall not require time-bound or just-in-time privilege elevation unless such capability is part of the declared CRS product configuration and is required by the corresponding requirement.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CRS fine-grained, policy-driven, or context-aware administrative authorization model.

- Documentation identifying separation-of-duties rules, contextual policy inputs, policy decision points, or equivalent mechanisms supported by the CRS, where applicable.
- Documentation identifying whether integration with external identity, access-management, or policy decision systems is part of the declared product configuration.
- A test configuration with representative administrative identities, permissions, contextual attributes, policy inputs, or external authorization integration, as applicable to the declared product configuration.

4. Assessment activities
The assessment shall at least:

1) Review the documented fine-grained, policy-driven, or context-aware CRS administrative authorization model.
2) Verify that the CRS enforces the applicable authorization policy for representative administrative operations.
3) Where separation of duties is supported or required by the declared configuration, verify that conflicting or incompatible administrative operations are not permitted to the same identity unless explicitly authorized by policy.
4) Where context-aware enforcement is supported or applicable, verify that authorization decisions take the documented contextual inputs into account.
5) Where integration with external identity, access-management, or policy decision systems is part of the declared product configuration, verify that authorization decisions use the integrated system as documented.

5. Assessment verdict

- Pass: the CRS enforces fine-grained, policy-driven, or context-aware authorization for administrative operations, including separation of duties, contextual policy evaluation, or external authorization integration where applicable to the requirement and declared product configuration.
- Fail: the CRS does not enforce fine-grained, policy-driven, or context-aware administrative authorization where required; or separation-of-duties, contextual policy evaluation, or external authorization integration is claimed but not implemented or not effective in the declared product configuration.

6. Assessment evidence

- [DESIGN] Documentation of the fine-grained, policy-driven, or context-aware CRS administrative authorization model.
- [CONFIG] Authorization policy, contextual policy, separation-of-duties configuration, or external authorization integration configuration, as applicable.
- [LOG] / [OBSERVATION] Evidence that administrative authorization decisions are enforced according to the documented policy and declared product configuration.

6.4.1.11 Assessment for Service Authorization

6.4.1.11.1 Assessment Case AC-CRS-SERV-AUTHZ-001

1. Assessment reference
Requirement: REQ-CRS-SERV-AUTHZ-001 (Basic)

2. Assessment objective
Verify that:

1) A default-deny policy is enforced for all CRS service accounts.
2) Service accounts are only permitted to perform actions explicitly authorized by CRS configuration or policy.

3. Assessment preparation
The assessment shall have access to:

- List of CRS service accounts and service-facing interfaces.
- Documentation of the authorization model and default behaviour for service accounts.
- At least one configuration with several service accounts, including one with no explicit permissions.

4. Assessment activities
The assessment shall at least:

1) Confirm from documentation/configuration that the default behaviour for service accounts is to deny any action not explicitly authorized.
2) Verify that a service account with no explicit permissions cannot perform management or data-related operations.
3) For a service account with defined permissions, verify that actions within its configured permissions are allowed and that non-authorized actions are denied.

5. Assessment verdict

- Pass: default behaviour for service accounts is deny; unprivileged service accounts cannot perform operations; and only explicitly authorized actions are permitted.
- Fail: unspecified actions are allowed for service accounts; or a service account without explicit permissions can perform operations.

6. Assessment evidence

- [SCOPE] List of CRS service accounts and interfaces in scope.
- [CONFIG] Service-account authorization and default policy configuration.
- [DESIGN] Description of the CRS service-account authorization model and default-deny behaviour.
- [LOG] / [OBSERVATION] Evidence of denied operations for unprivileged service accounts and successful execution of explicitly authorized actions.

6.4.1.11.2 Assessment Case AC-CRS-SERV-AUTHZ-002

1. Assessment reference
Requirement: REQ-CRS-SERV-AUTHZ-002 (Elevated)

2. Assessment objective
Verify that:

1) Granular authorization is enforced for CRS service accounts, with permitted actions bound to the authenticated service identity.
2) Least-privilege permissions are enforced for each service account.

3. Assessment preparation
The assessment shall have access to:

- Documentation on how service identities are represented and bound to permissions.
- Documentation on the permission or role model for CRS service accounts.
- At least one configuration with several service accounts having different, clearly scoped permissions.

4. Assessment activities
The assessment shall at least:

1) Confirm that authorization decisions for service accounts are based on the authenticated service identity, not solely on generic network or system properties.

2) For at least two service accounts with different permissions, verify that each account can perform only the actions explicitly bound to its identity and that actions belonging to another service identity are denied.

3) Verify that a service account configured with a minimal permission set cannot perform operations outside that defined set, demonstrating least privilege.

5. Assessment verdict

- Pass: authorization for service accounts is identity-bound and granular, and least-privilege permission sets can be configured and are enforced for individual service accounts.
- Fail: actions are not clearly bound to service identity; or service accounts can perform operations beyond their assigned permissions; or least-privilege cannot be demonstrated.

6. Assessment evidence

- [CONFIG] Service-account authorization configuration showing bindings between specific identities and allowed actions, including at least one minimal-permission account.
- [DESIGN] Documentation of the CRS service-account authorization model, identity binding and least-privilege approach.
- [LOG] / [OBSERVATION] Evidence of allowed operations within each service account's permissions and denied operations outside that scope.

6.4.1.11.3 Assessment Case AC-CRS-SERV-AUTHZ-003

1. Assessment reference
Requirement: REQ-CRS-SERV-AUTHZ-003 (Advanced)

2. Assessment objective
Verify that:

1) Fine-grained, policy-driven authorization is enforced for CRS service accounts.

2) Permissions for service accounts are bound to cryptographic service identities.

3) Context-aware enforcement (e.g. namespace, cluster, labels or similar attributes) is supported for service-account authorization decisions.

4) Integration with external authorization systems is supported, where available, to enable centralized management and evaluation of service-account policies.

3. Assessment preparation
The assessment shall have access to:

- Documentation of policy-driven authorization for CRS service accounts (policy language/rules, conditions, effects).
- Documentation on how cryptographic service identities (e.g. certificate identities, token claims) are bound to permissions.
- Documentation on context attributes available to policy evaluation (e.g. environment, namespace, workload labels).
- Documentation on integration with external authorization systems (PDPs/policy engines), if supported.
- At least one configuration where:
  - service-account policies reference cryptographic identities and at least one context attribute; and
  - where supported, an external authorization system is integrated and used for policy evaluation.

4. Assessment activities
The assessment shall at least:

1) Verify that CRS service-account authorization can be expressed and enforced via fine-grained policies and that permissions for at least one service account are defined in such policies.

2) Verify that policies can reference cryptographic identifiers of service accounts and that, in the assessed configuration, authorization decisions for at least one service depend on its cryptographic identity.

3) Verify that at least one context attribute (e.g. namespace, cluster, label) influences authorization decisions for service accounts and that access is allowed or denied based on that context in the assessed configuration.

4) Where integration with an external authorization system is supported, verify that the CRS consults or delegates service-account authorization decisions to that system and that policy changes in the external system are reflected in CRS authorization outcomes.

5. Assessment verdict

- Pass: service-account authorization is fine-grained and policy-driven; permissions are bound to cryptographic service identities; context-aware enforcement is supported; and, where applicable, external authorization integration allows centralized policy management and evaluation.

- Fail: policies cannot express fine-grained service-account authorization; or permissions are not effectively bound to cryptographic identities; or context attributes cannot influence authorization decisions; or supported external authorization integration is not effective.

6. Assessment evidence

- [CONFIG] Policy configuration for service-account authorization referencing cryptographic identities and context attributes, and, where applicable, external authorization integration settings.

- [DESIGN] Documentation of the policy-driven CRS service authorization architecture, including cryptographic identity binding, context handling and external authorization interfaces.

- [LOG] / [OBSERVATION] Evidence of authorization decisions depending on cryptographic identity and context, and of changes resulting from external authorization policy updates where supported.

6.4.1.12 Assessment for Confidentiality Protection

6.4.1.12.1 Assessment Case AC-CRS-CONF-001

1. Assessment reference
Requirement: REQ-CRS-CONF-001 (Basic)

2. Assessment objective
Verify that:

1) The CRS provides access-control mechanisms to restrict access to sensitive data under its control, including container images, configuration data, stored secrets, and security-relevant logs.

2) Access to such sensitive data is permitted only to authenticated and authorized entities through CRS-controlled interfaces and APIs, and not directly accessible to unauthenticated or unauthorized contexts (including container workloads).

3. Assessment preparation
The assessment shall have access to:

- Documentation describing:

  - the types of sensitive data handled by the CRS (for example container images under CRS control, configuration objects, secrets, security-relevant logs);

  - the storage locations and repositories where such data is kept;

- the CRS interfaces and APIs through which this data can be accessed or managed;
- the access-control model, including roles or identities authorized to access each type of sensitive data.

- A test configuration with:
  - at least one administrative account/role authorized to access and manage sensitive data;
  - at least one non-privileged or operational account/role with limited permissions;
  - one or more container workloads under CRS control;
  - CRS management interfaces and APIs reachable for testing.

4. Assessment activities

The assessment shall at least:

1) Review documentation and CRS configuration to identify:
   - which CRS repositories or objects correspond to container images, configuration data, stored secrets, and security-relevant logs;
   - which CRS-controlled interfaces and APIs are intended for accessing or managing each type of sensitive data;
   - which roles/accounts are authorized to use these interfaces.
2) Using an authenticated administrative account:
   - access representative examples of each sensitive data type via the documented CRS interfaces/APIs (for example listing and retrieving container images, viewing configuration objects, reading secrets via a CRS secret-management API, accessing security logs);
   - verify that such access is successful only via the CRS-controlled interfaces and behaves as documented.
3) Using a non-privileged account, unauthenticated context, or from within a container workload:
   - attempt to access or modify the same sensitive data via CRS APIs without appropriate permissions;
   - where underlying storage paths are visible, attempt direct access to data repositories outside of CRS-controlled interfaces;
   - verify that these attempts are denied or blocked and do not result in disclosure or modification of sensitive data.

5. Assessment verdict

- Pass: access to container images, configuration data, stored secrets, and security-relevant logs is restricted to authenticated and authorized entities via CRS-controlled interfaces; attempts by unauthenticated, unauthorized, or workload-level contexts to access or modify such data are prevented.
- Fail: sensitive data can be read or modified without using CRS-controlled interfaces, or by unauthenticated or unauthorized contexts; or the implemented access-control behaviour differs from the documented model.

6. Assessment evidence

- [DESIGN] Documentation of sensitive data types, storage locations, CRS interfaces/APIs, and the access-control model.
- [CONFIG] CRS configuration showing role/permission assignments and access-control settings for sensitive data.
- [LOG] Audit or event logs capturing successful authorized access and rejected unauthorized attempts, where such logging is implemented.
- [OBSERVATION] Test execution records (CLI/API outputs, management UI screenshots) demonstrating authorized access succeeding and unauthorized access failing.

- [SCOPE] Evidence identifying which CRS components and data stores were considered in-scope for sensitive data.

6.4.1.12.2 Assessment Case AC-CRS-CONF-002

1. Assessment reference
Requirement: REQ-CRS-CONF-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS supports encryption of sensitive data at rest, including container images, configuration data, stored secrets, and security-relevant logs, either via native capabilities or by integrating with encrypted storage, registries, or secret-management services.

2) When configured, encryption is effectively applied to the relevant sensitive data repositories.

3) Encryption keys used for data-at-rest protection are stored and managed in a manner that prevents access by container workloads and other unprivileged components.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing:
  - the CRS model for encrypting sensitive data at rest (native encryption, use of encrypted storage/registries, integration with secret-management services);
  - which data types and repositories are covered;
  - supported key-management approaches (for example internal key store, external KMS, platform services);
  - how encryption and key management are configured and monitored;
  - restrictions that prevent container workloads and other unprivileged components from accessing keys.
- A test configuration with:
  - one or more CRS data stores (for example image registry, configuration store, secret store, log store) configured to use the documented data-at-rest protection mechanisms;
  - an administrative account authorized to configure and monitor encryption and key management;
  - one or more container workloads and a non-privileged host context for attempting access to encrypted data and keys;
  - where applicable, access to underlying storage or diagnostic tools to observe whether data is stored encrypted.

4. Assessment activities
The assessment shall at least:

1) Review documentation and configuration to identify:
   - which sensitive data categories and storage locations are expected to be protected at rest;
   - how encryption is enabled and verified for each type (for example flags in CRS configuration, storage-layer encryption indicators);
   - where encryption keys are stored and how access to them is restricted.
2) With data-at-rest protection enabled according to product guidance:
   - create or update representative container images, configuration objects, stored secrets, and security logs;

- use platform or storage diagnostics (for example tools showing block- or file-level contents, provider encryption status APIs) to verify that these items are stored in encrypted form and not in recoverable plaintext;

- verify that disabling or revoking access to encryption keys prevents normal decryption and access to the protected data as expected.

3) From container workloads and other non-privileged contexts:

- attempt to access encryption keys or key stores via CRS APIs, environment variables, mounted volumes, or host-level resources;

- verify that such attempts are denied and that key material is not readable or exportable by these contexts.

4) Where external or platform key-management services are used:

- confirm via their interfaces that keys used for CRS data-at-rest protection reside in those services;

- verify that the CRS only obtains derived or wrapped keys as documented, and does not expose raw key material to workloads.

5. Assessment verdict

- Pass: the CRS supports and correctly applies data-at-rest protection to the configured categories of sensitive data; protected data is not stored in plaintext on underlying storage; and encryption keys are stored and managed in a way that prevents access by container workloads and other unprivileged components.

- Fail: sensitive data configured to be protected at rest is stored unencrypted or in recoverable plaintext; keys are accessible to workloads or non-privileged contexts; or behaviour diverges from the documented data-at-rest and key-management model.

6. Assessment evidence

- [DESIGN] Documentation of the CRS data-at-rest encryption model, including covered data types and key-management design.

- [CONFIG] CRS configuration showing enabled encryption settings for image, configuration, secret, and log stores, and key-management configuration.

- [LOG] Events or audit trails related to enabling/disabling encryption, key rotation, and failed access attempts to protected data or keys.

- [OBSERVATION] Test records and storage or provider-tool outputs demonstrating that data is stored encrypted and is inaccessible when keys are revoked.

- [PLT-CAP] Evidence of platform, storage, registry, or secret-management capabilities used to provide data-at-rest encryption.

6.4.1.12.3 Assessment Case AC-CRS-CONF-003

1. Assessment reference
Requirement: REQ-CRS-CONF-003 (Elevated)

2. Assessment objective
Verify that:

1) The CRS supports the use of encrypted communication for CRS management and control interfaces, for communication with image registries, and for other CRS APIs that handle sensitive data or control operations.

2) When these communications are configured to use encrypted transport, network traffic is protected with appropriate cryptographic mechanisms that ensure confidentiality, integrity, and peer authentication.

3) The CRS prevents fallback or downgrade from configured encrypted transport to unencrypted or weakly protected channels.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing:
  - CRS management and control interfaces and APIs that handle sensitive data or control operations;
  - communication paths between the CRS and image registries, secret stores, or other services handling sensitive data;
  - supported encrypted transport mechanisms (for example TLS configurations, integration with service meshes or platform-provided secure channels);
  - how anti-fallback or anti-downgrade behaviour is implemented and configured.
- A test configuration with:
  - CRS management and control interfaces reachable over the network;
  - a configured image registry and any other CRS APIs that handle sensitive data;
  - valid certificates or other credentials for establishing encrypted channels as documented;
  - tools to capture and inspect network traffic on relevant links.

4. Assessment activities
The assessment shall at least:

1) Review documentation and CRS configuration to identify:
   - which CRS interfaces and API endpoints are in scope for encrypted communication;
   - the recommended or required secure transport protocols and cipher configurations;
   - any configuration settings that enforce or relax requirements for encryption.
2) Configure encrypted transport for:
   - at least one CRS management or control interface;
   - communication between the CRS and a configured image registry;
   - - at least one additional CRS API that handles sensitive data or control operations.

   For each case, initiate typical management, registry, or API operations while capturing network traffic, and verify that:
   - payload data is encrypted and not visible in clear text on the wire;
   - negotiated cryptographic parameters conform to the configuration and are not weaker than documented.
3) Attempt to force downgrade or fallback by:
   - misconfiguring clients to request unencrypted protocols;
   - proposing weak or deprecated cipher suites if the protocol allows negotiation;
   - modifying CRS configuration, where possible, to request or allow unencrypted access despite a requirement for encryption.

   Verify that, when encrypted transport is mandated, the CRS does not silently fall back to unencrypted or weakly protected channels and instead either maintains encrypted communication or fails the connection in a documented way.

5. Assessment verdict

- Pass: CRS management/control interfaces, registry communication, and relevant APIs can be configured to use encrypted transport; captured traffic for these configured channels is encrypted and uses acceptable cryptographic parameters; and attempts to downgrade to unencrypted or weakly protected channels are prevented or cause connection failure as documented.
- Fail: encrypted transport cannot be enabled as documented; traffic configured to be encrypted is observed in plaintext or with weaker protection than configured; or the CRS silently falls back to unencrypted or weakly protected channels under downgrade attempts.

6. Assessment evidence

- [DESIGN] Documentation of secure communication mechanisms for CRS management, registry access, and sensitive APIs, including downgrade-prevention behaviour.
- [CONFIG] CRS configuration showing enabled encrypted transport settings and certificate or credential configuration.
- [LOG] Events or audit records relating to secure channel establishment, certificate or key errors, and rejected downgraded connections.
- [OBSERVATION] Packet captures and analysis outputs demonstrating encrypted traffic for configured channels and failed or rejected downgrade attempts.
- [PLT-CAP] Evidence of platform or service-mesh capabilities used to provide secure communication, where applicable.

6.4.1.12.4 Assessment Case AC-CRS-CONF-004

1. Assessment reference
Requirement: REQ-CRS-CONF-004 (Advanced)

2. Assessment objective
Verify that, where the underlying platform provides mechanisms for isolated or protected execution environments:

1) The CRS supports provisioning and management of execution environments for containerized workloads in which confidential data associated with those workloads is protected, during execution, from observation, inspection, or access through the CRS, the host operating system, and other workloads.
2) The CRS provides configuration and API interfaces to deploy workloads into such protected execution environments.
3) The CRS supports secure provisioning of secrets and key material to those workloads such that the secrets and key material are not exposed outside the protected provisioning path to the host operating system, other workloads, or CRS components not authorized to handle those materials.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing CRS support for platform-provided confidential or protected execution environments.
- Documentation describing CRS configuration and API interfaces used to deploy workloads into such protected execution environments.
- Documentation describing the secret and key provisioning path used for protected workloads.
- Documentation identifying CRS components authorized to handle secrets or key material during protected provisioning, and CRS components not authorized to handle those materials.
- Documentation identifying any dependency on platform mechanisms, firmware, Host OS services, or external trust services outside the declared product boundary, together with expected security properties.

- A test environment with at least one containerized workload deployable into a confidential or protected execution environment, where such environments are supported by the declared product configuration.

- Tools or interfaces available in the assessed environment to attempt inspection of confidential memory, secrets, or key material through CRS, host, or workload-level interfaces.

4. Assessment activities

The assessment shall at least:

1) Review the documented CRS protected-execution model and identify the security properties claimed for protected containerized workloads.

2) Review the documented behaviour of non-confidential containerized workloads only where such comparison is relevant to the product's protected-execution model. This comparison shall not be required where the product does not support an equivalent non-confidential mode or where such inspection capability is not provided by the declared product.

3) Deploy the workload into a confidential or protected execution environment and:

- verify that the workload is identified as running in the protected execution environment according to the product documentation;

- attempt, using available CRS, host, or workload-level interfaces within the assessed environment, to observe, inspect, or access confidential workload memory, process state, secrets, or key material;

- verify that direct inspection of confidential memory or secrets through CRS, host, or workload-level interfaces available in the assessed environment is blocked, or that retrieved data is protected or unintelligible, in accordance with the documented protected-execution model.

4) For secrets and key material provisioned to the protected workload:

- verify that the provisioning path is documented;

- identify CRS components authorized to handle the secrets or key material;

- attempt to access the secrets or key material through the host operating system, other workloads, and CRS components or interfaces not authorized to handle those materials;

- verify that the secrets or key material are not exposed outside the protected provisioning path.

5. Assessment verdict

- Pass: the CRS can deploy workloads into platform-provided confidential or protected execution environments; in such environments, confidential workload data in use cannot be observed, inspected, or accessed through CRS, host operating system, or workload-level interfaces beyond what is documented in the protected-execution model; and secrets and key material are not exposed outside the protected provisioning path to the host operating system, other workloads, or CRS components not authorized to handle those materials.

- Fail: workloads deployed as confidential or protected can have confidential data observed, inspected, or accessed in clear form through CRS, host operating system, or workload-level interfaces contrary to the documented protected-execution model; or secrets or key material are exposed outside the protected provisioning path to the host operating system, other workloads, or CRS components not authorized to handle those materials; or dependencies outside the declared product boundary are not identified.

6. Assessment evidence

- [DESIGN] Documentation of the CRS integration with platform confidential or protected execution mechanisms.

- [DESIGN] Documentation of the secret and key provisioning model, including the protected provisioning path and CRS components authorized to handle secrets or key material.

- [CONFIG] CRS and platform configuration showing how confidential or protected workloads are enabled and identified.

- [OE] Documentation identifying dependencies on platform mechanisms, firmware, Host OS services, or external trust services outside the declared product boundary, including expected security properties.
- [LOG] / [OBSERVATION] Evidence of workload deployment into the protected execution environment.
- [OBSERVATION] Evidence that attempts to inspect confidential memory or secrets through CRS, host, or workload-level interfaces are blocked or result only in protected or unintelligible data.
- [OBSERVATION] Evidence that secrets and key material are not exposed outside the protected provisioning path to the host operating system, other workloads, or CRS components not authorized to handle those materials.
- [PLT-CAP] Evidence of platform confidential-computing or protected-execution capabilities used by the CRS, where applicable.

6.4.1.13 Assessment for Availability and Resilience

6.4.1.13.1 Assessment Case AC-CRS-AVAIL-001

1. Assessment reference
Requirement: REQ-CRS-AVAIL-001 (Basic)

2. Assessment objective
Verify that:

1) the CRS maintains resource isolation between container workloads and protects CRS control-plane processes from resource exhaustion caused by container workloads, covering at least CPU, memory, and storage resources;
2) failure or abnormal termination of an individual container is contained within its execution context and, under normal operating conditions and within configured resource limits, does not compromise operation of the CRS control plane or cause other containers to terminate unexpectedly.

3. Assessment preparation
The assessment shall have access to:

- Documentation of CRS resource isolation mechanisms and limits for CPU, memory, and storage resources, and how these mechanisms protect CRS control-plane processes.
- Documentation of how container failure or abnormal termination is contained and what behaviour is expected for other containers and CRS control-plane components under normal operating conditions and configured resource limits.
- A test configuration with multiple containers, including at least one container capable of generating high CPU, memory, and storage load and one or more normal workloads.

4. Assessment activities
The assessment shall at least:

1) Confirm that CPU, memory, and storage isolation or limitation mechanisms can be configured for container workloads and that CRS control-plane processes are protected from arbitrary workload exhaustion of those resources.
2) Generate high CPU, memory, and storage load from a container workload within the assessed configuration and verify that CRS control-plane functions remain operational and responsive.
3) Induce abnormal termination of a selected container and verify that the failure is contained within that container execution context and does not cause unexpected termination of other containers or compromise CRS control-plane operation under normal operating conditions and configured resource limits.

5. Assessment verdict

- Pass: CPU, memory, and storage isolation mechanisms protect CRS control-plane processes from workload-induced exhaustion; and failure of an individual container is contained within its execution context without compromising CRS control-plane operation or causing unexpected termination of other containers under normal conditions and configured limits.
- Fail: CRS control-plane processes are not adequately protected from workload-induced exhaustion; or failure of an individual container propagates beyond its execution context and compromises the CRS control plane or other containers under normal conditions and configured limits.

6. Assessment evidence

- [CONFIG] Resource isolation and limit configuration for CPU, memory, and storage resources for containers and control-plane components.
- [DESIGN] Documentation of CRS resource isolation and failure-containment architecture.
- [LOG] / [OBSERVATION] Evidence showing control-plane availability under CPU, memory, and storage stress and containment of individual container failure.

6.4.1.13.2 Assessment Case AC-CRS-AVAIL-002

1. Assessment reference
Requirement: REQ-CRS-AVAIL-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS provides automatic recovery mechanisms to restart failed containers based on defined policies, without manual intervention.
2) Where supported by the deployment environment, these mechanisms can be used with orchestration or clustering systems to restore workloads on the same or an alternative host.

3. Assessment preparation
The assessment shall have access to:

- Documentation of CRS container restart policies and automatic recovery behaviour (e.g. "always", "on-failure", back-off behaviour).
- Documentation of integration with orchestration or clustering systems, where supported (e.g. Kubernetes®, Swarm, other orchestrators).
- A test environment with several containers, and where supported, at least two hosts under common orchestration/cluster control.

4. Assessment activities
The assessment shall at least:

1) Configure one or more containers with automatic restart policies as described in documentation.
2) Induce container failure (e.g. crash, forced termination) and verify that the CRS automatically restarts the container according to the defined policy, without manual actions.
3) Where orchestration/clustering is supported, simulate host or node unavailability for a container workload and verify that the orchestration/cluster in conjunction with CRS mechanisms restores the workload on the same or an alternative host according to the configuration.

5. Assessment verdict

- Pass: failed containers configured with restart policies are automatically restarted without manual intervention; and, where supported, orchestration/clustering combined with CRS behaviour restores workloads on the same or another host as documented.

- Fail: container restart requires manual intervention in contradiction with configured policies; or supported orchestration/clustering cannot restore workloads as intended.

6. Assessment evidence

- [CONFIG] Container restart policy configuration and, where applicable, orchestration/cluster configuration.
- [DESIGN] Documentation of CRS automatic recovery mechanisms and integrations with orchestration/clustering systems.
- [LOG] / [OBSERVATION] Evidence of automatic container restarts and, where applicable, restoration of workloads on alternative hosts.

6.4.1.13.3 Assessment Case AC-CRS-AVAIL-003

1. Assessment reference
Requirement: REQ-CRS-AVAIL-003 (Elevated)

2. Assessment objective
Verify that:

1) The CRS implements resource scheduling or reservation mechanisms that allow configuration of minimum resource allocations for CRS security and management functions (CPU, memory and I/O).

2) These mechanisms protect the CRS control plane from resource starvation caused by container workloads or external contention, ensuring that security and management functions remain able to operate under load.

3. Assessment preparation
The assessment shall have access to:

- Documentation of CRS resource scheduling/reservation or priority mechanisms (e.g. control-plane reservations, priorities, QoS classes).
- Documentation describing how minimum CPU, memory and I/O resources can be reserved or prioritized for CRS security and management components.
- A test configuration with CRS security/management functions active and multiple container workloads capable of generating high CPU, memory and I/O load.

4. Assessment activities
The assessment shall at least:

1) Confirm that resource scheduling or reservation mechanisms exist and that minimum CPU, memory and I/O allocations (or equivalent prioritization) can be configured for CRS security and management functions.

2) Configure such minimum allocations/priorities in the assessed environment.

3) Generate high resource load from containers and/or external sources (CPU- and I/O-intensive workloads) and verify that:

- CRS management and security interfaces remain responsive; and
- critical control-plane operations (e.g. container lifecycle operations, logging, policy enforcement) continue to function correctly under load.

5. Assessment verdict

- Pass: resource scheduling/reservation mechanisms support configuration of minimum CPU, memory and I/O allocations (or equivalent prioritization) for CRS security and management functions; and under high load from container workloads or external contention, these functions remain able to operate as intended.
- Fail: minimum allocations/priorities for security/management functions cannot be configured; or under load, the CRS control plane or security/management functions become starved and unable to operate.

6. Assessment evidence

- [CONFIG] Resource scheduling/reservation or prioritization configuration for CRS security and management components.
- [DESIGN] Documentation of CRS resource scheduling architecture and its application to control-plane protection.
- [LOG] / [OBSERVATION] Evidence that CRS management and security functions continue to operate correctly under stress conditions.

6.4.1.13.4 Assessment Case AC-CRS-AVAIL-004

1. Assessment reference
Requirement: REQ-CRS-AVAIL-004 (Advanced)

2. Assessment objective
Verify that, where the deployment environment supports workload relocation or continuity-preserving recovery, the CRS supports mechanisms to reduce interruption of selected container workloads following control-plane or host-level failures, such as restart from a trusted state, integration with orchestrator recovery, checkpoint and restore where supported, runtime-state preservation, or equivalent continuity mechanisms.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing CRS-supported continuity mechanisms for selected container workloads, such as restart from a trusted state, integration with orchestrator recovery, checkpoint and restore where supported, runtime-state preservation, or equivalent continuity mechanisms.
- Documentation describing the scope, limitations, required environment support, and expected recovery or interruption behaviour of the supported continuity mechanisms.
- Documentation identifying any orchestration, clustering, storage, networking, platform, or external lifecycle-management dependencies required for the continuity mechanism, together with the expected security properties.
- A test environment where workload relocation or continuity-preserving recovery is supported, and at least one container workload configured to use one of the documented continuity mechanisms.

4. Assessment activities
The assessment shall at least:

1) Identify the CRS continuity mechanism supported for selected container workloads in the assessed configuration.
2) Configure a selected container workload to use the documented continuity mechanism, including any required integration with orchestration, clustering, storage, networking, platform, or external lifecycle-management mechanisms.
3) Simulate a control-plane or host-level failure, or initiate a planned relocation or recovery action, according to the documented mechanism.
4) Verify that the selected workload is restarted, recovered, relocated, restored, or otherwise continued according to the documented continuity mechanism.
5) Observe workload behaviour and confirm that disruption is reduced in accordance with the documented mechanism and expected recovery behaviour.
6) Verify that any orchestration, clustering, storage, networking, platform, or external lifecycle-management dependency required by the continuity mechanism is documented together with expected security properties.

5. Assessment verdict

- Pass: the CRS supports at least one documented continuity mechanism for selected container workloads, such as restart from a trusted state, integration with orchestrator recovery, checkpoint and restore where supported, runtime-state preservation, or equivalent continuity mechanism; the selected mechanism operates as documented following a control-plane or host-level failure, planned relocation, or recovery action; workload interruption is reduced in accordance with the documented recovery behaviour; and required environmental dependencies are documented with expected security properties.

- Fail: the CRS does not support a continuity mechanism required by REQ-CRS-AVAIL-004 in the assessed configuration; or the selected mechanism does not operate as documented; or workload interruption is not reduced in accordance with the documented recovery behaviour; or required environmental dependencies are not documented.

6. Assessment evidence

- [CONFIG] Configuration of the selected continuity mechanism for selected container workloads, including any orchestrator, cluster, storage, networking, platform, or external lifecycle-management settings.

- [DESIGN] Documentation of CRS continuity architecture, including supported mechanisms, scope of preserved or recovered state where applicable, limitations, dependencies, and expected recovery behaviour.

- LOG] / [OBSERVATION] Evidence of restart, recovery, relocation, restore, checkpoint/restore where supported, runtime-state preservation, or equivalent continuity operation during and after the simulated failure, planned relocation, or recovery action.

- [OE] Documentation identifying environmental dependencies and expected security properties for the continuity mechanism, where applicable.

6.4.1.14 Assessment for Logging

6.4.1.14.1 Assessment Case AC-CRS-LOG-001

1. Assessment reference
Requirement: REQ-CRS-LOG-001 (Basic)

2. Assessment objective
Verify that:

1) The CRS generates audit logs for security-relevant administrative actions and security events, including at least:

- authentication attempts to CRS administrative or control interfaces;

- configuration changes affecting CRS behaviour or security posture;

- container lifecycle operations (create, start, stop, delete, restart) initiated through the CRS.

2) Audit logs are not writable or directly modifiable by container workloads or other unprivileged processes running within container namespaces.

3. Assessment preparation
The assessment shall have access to:

- Documentation of CRS logging/auditing capabilities and log storage locations.

- Documentation of which actions/events are treated as security-relevant.

- A configuration with:

  - CRS administrative/control access enabled;

  - at least one container workload that can be created, started, stopped, restarted and deleted.

4. Assessment activities
The assessment shall at least:

1) Identify audit log sources and storage locations from documentation/configuration.

2) In the assessed configuration, perform:

- successful and failed authentication attempts to CRS administrative/control interfaces;
- CRS configuration changes affecting behaviour/security posture;
- container lifecycle operations (creation, start, stop, restart, deletion) via CRS mechanisms.

Verify that corresponding audit records are generated, including at least action/event type, affected object, timestamp and responsible account or process where applicable.

3) From a container workload and from another unprivileged process, attempt to write to, modify or delete the CRS audit log files or endpoints and verify that these operations are denied.

5. Assessment verdict

- Pass: the CRS generates audit logs for the specified security-relevant actions/events; and audit logs cannot be written or directly modified by container workloads or other unprivileged processes in container namespaces.
- Fail: one or more required events are not logged; or container workloads/unprivileged processes can write or modify audit logs.

6. Assessment evidence

- [CONFIG] CRS logging configuration and log storage settings.
- [DESIGN] Documentation of audit logging behaviour, logged event categories and log isolation from workloads.
- [LOG] / [OBSERVATION] Sample audit entries for authentication attempts, configuration changes and container lifecycle operations, and evidence of denied write/modify attempts from containers/unprivileged processes.

6.4.1.14.2 Assessment Case AC-CRS-LOG-002

1. Assessment reference
Requirement: REQ-CRS-LOG-002 (Elevated)

2. Assessment objective
Verify that:

1) CRS audit logs are protected from unauthorized access, modification, or deletion;

2) access to CRS audit logs is restricted to authorized administrative roles or trusted system processes under the control of the CRS or host operating system;

3) the CRS supports secure export of audit logs to external log management or SIEM systems using authenticated and encrypted communication channels.

3. Assessment preparation
The assessment shall have access to:

- Documentation on audit log storage and access-control mechanisms.
- Documentation identifying which administrative roles and trusted system processes are allowed to access or manage CRS audit logs.
- Documentation of CRS log export mechanisms and supported authenticated and encrypted transport options.
- A configuration with audit logging enabled and a reachable external log management or SIEM endpoint that can accept logs over an authenticated and encrypted channel.

4. Assessment activities
The assessment shall at least:

1) Review configuration and underlying platform permissions to confirm that audit log storage can only be accessed by authorized administrative roles or trusted system processes and that modification or deletion by other users or processes is prevented.

2) Attempt to read, modify, or delete audit logs using non-authorized accounts or processes and verify that these attempts are denied.

3) Configure export of CRS audit logs to the external log or SIEM endpoint using an authenticated and encrypted communication channel as documented.

4) Verify, using configuration and where feasible protocol inspection, that the export uses authenticated and encrypted communication and that the external system receives log records corresponding to those generated locally.

5. Assessment verdict

- Pass: only authorized administrative roles or trusted system processes can access CRS audit logs; unauthorized access, modification, or deletion is prevented; and CRS audit logs can be securely exported to external log management or SIEM systems using authenticated and encrypted communication channels.

- Fail: unauthorized access, modification, or deletion of CRS audit logs is possible; authorized access restrictions are not enforced; or audit log export is not securely protected using authenticated and encrypted communication channels.

6. Assessment evidence

- [CONFIG] Audit log storage permissions, role or process mappings, and export configuration, including authentication and encryption settings.

- [DESIGN] Documentation of the CRS audit log protection model and external export mechanisms.

- [LOG] / [OBSERVATION] Evidence of denied access attempts by non-authorized entities and successful export to an external log or SIEM system over an authenticated and encrypted communication channel.

6.4.1.14.3 Assessment Case AC-CRS-LOG-003

1. Assessment reference
Requirement: REQ-CRS-LOG-003 (Advanced)

2. Assessment objective
Verify that:

1) the CRS supports cryptographic protection, trusted time-stamping, and tamper-evident protection of audit logs so that origin and integrity of audit records can be verified during forensic analysis;

2) where external time or signing services are used, audit records include sufficient metadata to identify and verify the trust source used for time-stamping and integrity protection.

3. Assessment preparation

The assessment shall have access to:

- Documentation describing cryptographic mechanisms used to protect CRS audit logs, including how tamper-evident protection is achieved and how verification is performed.

- Documentation of trusted time-stamping and, where applicable, external time or signing services, including the metadata recorded in audit logs about the trust source.

- A configuration with cryptographic log protection, trusted time-stamping, and tamper-evident protection enabled, with the ability to export or access raw audit log data and associated metadata for verification.

4. Assessment activities
The assessment shall at least:

1) Enable cryptographic protection, trusted time-stamping, and tamper-evident protection for CRS audit logs as documented.

2) Generate representative audit events and collect the corresponding audit records and associated metadata.

3) Using the documented verification tools or procedures, verify that cryptographic checks succeed, each audit record includes a trusted timestamp, and, where external time or signing services are used, the metadata clearly identifies the trust source and allows verification against that source.

4) Modify a copy of the audit log, for example by changing or removing records, and confirm that the verification process fails or indicates tampering.

5. Assessment verdict

- Pass: cryptographic protection, trusted time-stamping, and tamper-evident protection for CRS audit logs are available; audit records contain sufficient metadata to verify origin, integrity, and, where applicable, the trust source used for time-stamping and integrity protection; and tampering is detectable through the documented verification process.

- Fail: cryptographic protection, trusted time-stamping, or tamper-evident protection is absent or ineffective; audit records do not contain sufficient metadata to verify the trust source where applicable; or tampering cannot be reliably detected through the documented verification process.

6. Assessment evidence

- [CONFIG] Configuration enabling cryptographic protection, trusted time-stamping, and tamper-evident protection of CRS audit logs, including any external signing or time-service integration.

- [DESIGN] Documentation of CRS log protection and time-stamping architecture, including metadata format and verification procedures.

- [LOG] / [OBSERVATION] Audit logs and verification outputs showing successful checks for unmodified logs and detection of tampering or invalid trust-source conditions.

6.4.1.15 Assessment for Secure Update

6.4.1.15.1 Assessment Case AC-CRS-UPD-001

1. Assessment reference
Requirement: REQ-CRS-UPD-001 (Basic)

2. Assessment objective
Verify that:

1) the CRS is delivered in a form that supports authenticity and integrity verification of CRS updates before installation or activation;

2) the CRS supports authenticity and integrity verification of CRS updates using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata;

3) Where the CRS includes the mechanism that verifies, installs, or activates its updates, this mechanism prevents installation or activation of an update when authenticity or integrity verification fails and records a security-relevant event;

4) Where update delivery, verification, installation, or activation is performed by a Host OS update service, package manager, platform update service, appliance update mechanism, CE, CO, orchestration-based update pipeline, or other Operational Environment mechanism outside the declared CRS product or outside the responsibility of the manufacturer, the product security documentation identifies that mechanism, the expected security properties, and the assumptions required for secure update of the CRS.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying the CRS components delivered as part of the declared CRS product.
- Documentation describing the update artefacts applicable to those CRS components.
- Documentation describing how authenticity and integrity verification of CRS updates is supported before installation or activation, including the verification metadata used, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Documentation identifying whether update delivery, verification, installation, or activation is performed by the CRS, by another component under the responsibility of the manufacturer, or by an Operational Environment mechanism.
- Where the CRS includes the mechanism that verifies, installs, or activates updates, a test configuration allowing use of a valid update artefact and an update artefact with invalid, missing, revoked, or intentionally modified authenticity or integrity protection, or an equivalent test condition.
- Where update delivery, verification, installation, or activation is performed by an Operational Environment mechanism, product security documentation identifying the expected update mechanism, expected security properties, and assumptions required for secure update of the CRS.

4. Assessment activities
The assessment shall at least:

1) Identify the CRS components and update artefacts covered by REQ-CRS-UPD-001.
2) Check that the CRS is delivered in a form that supports authenticity and integrity verification of CRS updates before installation or activation.
3) Check that the update artefacts support authenticity and integrity verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
4) Check whether update delivery, verification, installation, or activation is performed by the CRS, by another component under the responsibility of the manufacturer, or by an Operational Environment mechanism.
5) Where the CRS includes the mechanism that verifies, installs, or activates updates, attempt to install or activate an update whose authenticity or integrity verification fails and verify that installation or activation is prevented and that a security-relevant event is recorded.
6) Where update delivery, verification, installation, or activation is performed outside the declared CRS product or outside the responsibility of the manufacturer, verify that the product security documentation identifies the Operational Environment dependency, the expected security properties, and the assumptions required for secure update of the CRS.

5. Assessment verdict

- Pass: the CRS is delivered in a form that supports authenticity and integrity verification of CRS updates before installation or activation; update artefacts support verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata; and, where the CRS includes the mechanism that verifies, installs, or activates updates, failed authenticity or integrity verification prevents installation or activation and is recorded as a security-relevant event. Where update delivery, verification, installation, or activation is performed outside the declared CRS product or outside the responsibility of the manufacturer, the Operational Environment dependency, expected security properties, and assumptions are documented.
- Fail: the CRS is not delivered in a form that supports authenticity and integrity verification of CRS updates before installation or activation; or the update artefacts do not support the required verification metadata; or, where the CRS includes the mechanism that verifies, installs, or activates updates, an update whose authenticity or integrity verification fails can be installed or activated, or the failure is not recorded as a security-relevant event; or the relevant Operational Environment dependency is not documented where update delivery, verification, installation, or activation is performed outside the declared CRS product or outside the responsibility of the manufacturer.

6. Assessment evidence

- [DESIGN] Documentation of the CRS update model, including update artefacts and the component or mechanism responsible for update delivery, verification, installation, or activation.
- [DESIGN] Documentation of authenticity and integrity verification metadata used for CRS updates, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- [CONFIG] Configuration of the CRS update mechanism, where the CRS includes the mechanism that verifies, installs, or activates updates.
- [LOG] / [OBSERVATION] Where the CRS includes the mechanism that verifies, installs, or activates updates, evidence that a valid update is accepted and that an update with failed authenticity or integrity verification is prevented from installation or activation and recorded as a security-relevant event.
- [GUIDANCE] / [OE] Where update delivery, verification, installation, or activation is performed outside the declared CRS product or outside the responsibility of the manufacturer, product security documentation identifying the Operational Environment dependency, expected security properties, and assumptions required for secure update of the CRS.

6.4.1.15.2 Assessment Case AC-CRS-UPD-002

1. Assessment reference
Requirement: REQ-CRS-UPD-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS supports rollback protection for CRS updates where rollback protection is provided by the CRS or by an update mechanism under the responsibility of the manufacturer.

2) Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed CRS updates is prevented.

3) Authorized rollback to a previously trusted CRS version is permitted only when:

a) the rollback image or package is verified for integrity and authenticity;

b) the rollback action is explicitly authorized by an administrator with appropriate privileges or an equivalent authorized control function; and

c) the rollback operation is recorded as a security-relevant event

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying whether rollback protection for CRS updates is provided by the CRS or by an update mechanism under the responsibility of the manufacturer.
- Documentation describing the rollback protection mechanism, where rollback protection is provided.
- Documentation describing procedures for authorized rollback to a previously trusted CRS version, where authorized rollback is supported.
- Documentation identifying the administrator privileges or equivalent authorized control function required to authorize rollback, where authorized rollback is supported.
- Documentation describing how rollback operations are recorded as security-relevant events, where rollback is supported.
- A test or demonstration configuration allowing an outdated, revoked, or replayed CRS update to be attempted, where rollback protection is provided by the CRS or by an update mechanism under the responsibility of the manufacturer.

- A test or demonstration configuration allowing authorized rollback to a previously trusted CRS version, where authorized rollback is supported.

4. Assessment activities
The assessment shall at least:

1) Check whether rollback protection for CRS updates is provided by the CRS or by an update mechanism under the responsibility of the manufacturer.

2) Where rollback protection is not provided by the CRS or by an update mechanism under the responsibility of the manufacturer, verify that the product security documentation identifies the update and rollback assumptions applicable to the declared product configuration.

3) Where rollback protection is provided, attempt to install an outdated, revoked, or replayed CRS update outside the documented authorized rollback procedure and verify that installation is prevented.

4) Where authorized rollback is supported, perform an authorized rollback using the documented procedure and verify that:

a) the rollback image or package is verified for integrity and authenticity;

b) the rollback action requires explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; and

c) the rollback operation is recorded as a security-relevant event.

5. Assessment verdict

- Pass: where rollback protection is provided by the CRS or by an update mechanism under the responsibility of the manufacturer, unauthorized installation of outdated, revoked, or replayed CRS updates is prevented; and authorized rollback to a previously trusted CRS version is permitted only following integrity and authenticity verification of the rollback image or package, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function, and recording of the rollback operation as a security-relevant event. Where rollback protection is not provided by the CRS or by an update mechanism under the responsibility of the manufacturer, the update and rollback assumptions applicable to the declared product configuration are documented.

- Fail: where rollback protection is provided by the CRS or by an update mechanism under the responsibility of the manufacturer, outdated, revoked, or replayed CRS updates can be installed without authorization; or authorized rollback can be performed without integrity and authenticity verification; or rollback can be performed without explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; or the rollback operation is not recorded as a security-relevant event.

6. Assessment evidence

- [DESIGN] Documentation identifying whether rollback protection is provided by the CRS or by an update mechanism under the responsibility of the manufacturer.

- [DESIGN] Documentation of rollback protection mechanisms and authorized rollback procedures, where rollback protection or authorized rollback is supported.

- [CONFIG] Configuration or policy information showing rollback authorization controls, where authorized rollback is supported.

- [LOG] / [OBSERVATION] Where rollback protection is provided, evidence that outdated, revoked, or replayed updates are prevented outside the authorized rollback procedure.

- [LOG] / [OBSERVATION] Where authorized rollback is supported, evidence that rollback is performed only after integrity and authenticity verification, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function, and recording as a security-relevant event.

- [GUIDANCE] / [OE] Where rollback protection is not provided by the CRS or by an update mechanism under the responsibility of the manufacturer, documentation identifying update and rollback assumptions applicable to the declared product configuration.

6.4.1.15.3 Assessment Case AC-CRS-UPD-003

1. Assessment reference
Requirement: REQ-CRS-UPD-003 (Advanced)

2. Assessment objective
Verify that:

1) the declared product configuration identifies whether coordinated update of CRS components across nodes is supported as part of the declared product.

2) where coordinated update of CRS components across nodes is supported as part of the declared product, the CRS provides, for each managed node, identification of the update action.

3) where coordinated update of CRS components across nodes is supported as part of the declared product, the CRS provides, for each managed node, verification of update outcome.

4) where coordinated update of CRS components across nodes is supported as part of the declared product, the CRS provides detection of failed or partial rollout states.

3. Assessment preparation
The assessment shall have access to:

- Product security documentation identifying whether coordinated update of CRS components across nodes is supported as part of the declared product.
- Documentation describing the coordinated update mechanism, where coordinated update is supported.
- Documentation describing how update actions are identified for each managed node, where coordinated update is supported.
- Documentation describing how update outcome is verified for each managed node, where coordinated update is supported.
- Documentation describing how failed or partial rollout states are detected, where coordinated update is supported.
- A representative test or demonstration environment with more than one managed node, only where coordinated update across nodes is supported as part of the declared product.

4. Assessment activities
The assessment shall at least:

1) Check whether coordinated update of CRS components across nodes is supported as part of the declared product.

2) If coordinated update of CRS components across nodes is not supported as part of the declared product, check that this is identified in the product security documentation and proceed to the verdict based on the condition in REQ-CRS-UPD-003 not being triggered.

3) Where coordinated update is supported, check that the CRS identifies the update action for each managed node.

4) Where coordinated update is supported, check that the CRS verifies the update outcome for each managed node.

5) Where coordinated update is supported, check that the CRS detects failed or partial rollout states.

5. Assessment verdict

- Pass: coordinated update of CRS components across nodes is not supported as part of the declared product and this is identified in the product security documentation.
- Pass: coordinated update of CRS components across nodes is supported as part of the declared product and the required evidence demonstrates that the CRS provides, for each managed node, identification of the update action, verification of update outcome, and detection of failed or partial rollout states.

- Fail: the product security documentation does not identify whether coordinated update of CRS components across nodes is supported as part of the declared product.
- Fail: coordinated update of CRS components across nodes is supported as part of the declared product and the required evidence is missing or does not demonstrate identification of the update action, verification of update outcome, or detection of failed or partial rollout states.

6. Assessment evidence

- [DESIGN]:
  - Product security documentation identifying whether coordinated update of CRS components across nodes is supported as part of the declared product.
  - Documentation of the coordinated update mechanism, where coordinated update is supported.
- [CONFIG] Configuration or policy documentation showing how managed nodes are identified for coordinated update, where coordinated update is supported.
- [LOG] / [OBSERVATION]:
  - Evidence showing identification of update actions per managed node, where coordinated update is supported.
  - Evidence showing update outcome verification per managed node, where coordinated update is supported.
  - Evidence showing detection of failed or partial rollout states, where coordinated update is supported.

6.4.1.16 Assessment for Secure Configuration and Default

6.4.1.16.1 Assessment Case AC-CRS-CFG-001

1. Assessment reference
Requirement: REQ-CRS-CFG-001 (Basic)

2. Assessment objective
Verify that:

1) By default, the CRS disables interfaces and features not required for container execution and CRS administration, including remote debugging endpoints, unauthenticated control sockets, and developer interfaces exposing management or control capabilities.
2) Only administrators can explicitly enable such interfaces and features through CRS configuration.

3. Assessment preparation
The assessment shall have access to:

- Documentation listing CRS interfaces/features (remote debugging, control sockets, developer interfaces, admin APIs/UIs) and identifying those required for normal container execution and CRS administration.
- Documentation of CRS configuration mechanisms and role/permission model for modifying configuration.
- A fresh/default CRS deployment in its documented default configuration, with at least one non-administrative account or context (e.g. unprivileged user, container workload).

4. Assessment activities
The assessment shall at least:

1) In the default configuration, attempt to access all documented non-essential interfaces/features (remote debugging endpoints, unauthenticated control sockets, developer interfaces) and verify they are disabled or not reachable.
2) Confirm that interfaces/features required for container execution and CRS administration are available as documented.

3) Using administrative credentials, enable one or more of the disabled interfaces/features via CRS configuration and verify that:

- the configuration change is accepted; and
- the corresponding interface/feature becomes available.

4) Attempt to perform the same enable operations using a non-administrative account/context and verify that the configuration change is rejected or not permitted.

5. Assessment verdict

- Pass: non-essential/debug/developer interfaces and unauthenticated control sockets are disabled by default; required interfaces for execution/administration are available; and only administrators can enable additional interfaces/features.
- Fail: non-essential or insecure interfaces are enabled by default; or non-administrative users can enable such interfaces/features.

6. Assessment evidence

- [CONFIG] Default and modified CRS configuration showing interface/feature states and permissions required for changes.
- [DESIGN] Documentation of CRS interfaces, default exposure, and configuration/role model.
- [OBSERVATION] / [LOG] Connection attempts and CLI/UI output showing:
  - non-availability of non-essential interfaces by default;
  - successful enable by admin;
  - failed enable attempts by non-admin.

6.4.1.16.2 Assessment Case AC-CRS-CFG-002

1. Assessment reference
Requirement: REQ-CRS-CFG-002 (Elevated)

2. Assessment objective
Verify that:

1) The CRS validates configuration parameters under its control before applying them.

2) Configuration validation rejects settings that would:

- disable or bypass mandatory security requirements defined in the present document; or
- compromise container-to-container isolation; or
- compromise controlled access to host resources.

3) When a configuration change is rejected for security reasons, the CRS provides clear feedback to the administrator indicating the reason.

3. Assessment preparation

The assessment shall have access to:

- Documentation of CRS configuration parameters and validation behaviour, including examples of security-relevant invalid settings (e.g. disabling required controls, removing isolation, granting unrestricted host access).
- Documentation of mandatory CRS security controls, isolation guarantees and host-access restrictions.
- A test environment where CRS configuration can be modified via supported administrative interfaces.

4. Assessment activities
The assessment shall at least:

1) Identify a set of configuration parameters where insecure values would:

- disable/bypass mandatory security controls (e.g. turning off mandatory auth/logging/admission checks);
- weaken or remove container-to-container isolation; or
- grant uncontrolled access to host resources that should remain restricted.

2) Attempt to apply such insecure configuration changes and verify that the CRS refuses to apply them as part of its validation logic.

3) Capture the feedback presented to the administrator and verify that it clearly indicates the security-related reason for rejection (e.g. reference to disabled control, isolation violation, or host-access violation).

4) Apply one or more valid configuration changes and verify they are accepted and enforced, demonstrating that validation focuses on insecure cases rather than blocking configuration in general.

5. Assessment verdict

- Pass: configuration parameters are validated before application; insecure settings that disable/bypass mandatory controls or compromise isolation/host access are rejected; and clear, security-related feedback is provided to the administrator.
- Fail: unsafe configurations can be applied; or validation does not cover key security/isolation/host-access guarantees; or error messages do not clearly explain the security reason.

6. Assessment evidence

- [CONFIG] Examples of attempted insecure and valid configuration changes and their outcomes.
- [DESIGN] Documentation of CRS configuration validation logic and its linkage to mandatory controls, isolation, and host-access rules.
- [LOG] / [OBSERVATION] CLI/UI output showing rejection of insecure configurations with explicit reasons, and acceptance of secure configurations.

6.4.1.16.3 Assessment Case AC-CRS-CFG-003

1. Assessment reference
Requirement: REQ-CRS-CFG-003 (Advanced)

2. Assessment objective
Verify that:

1) The CRS supports defining and using security configuration baselines that specify secure settings for:

- CRS daemon configuration; and
- default container security profiles, including at least isolation-related options and default access to host resources.

2) The CRS can automatically validate current CRS configuration and default container profile settings against a selected security baseline.

3) The CRS generates alerts or audit events when non-compliant settings are detected.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing security configuration baselines (how they are defined, stored, selected and scoped to daemon config and default container profiles).

- Documentation of baseline content, including isolation-related options (e.g. default capabilities, privilege levels, namespaces) and default host resource access (e.g. host volumes, host network).
- Documentation of automatic baseline validation mechanisms and alert/audit integration.
- A test environment where at least one security baseline can be defined/selected and CRS/default-profile settings can be altered to introduce non-compliance.

4. Assessment activities
The assessment shall at least:

1) Define or select a security baseline that includes secure settings for CRS daemon configuration and default container security profiles, covering isolation and host-access defaults as supported by the product.
2) Align CRS configuration and default container profiles with the baseline and invoke the baseline validation function; verify that the CRS reports compliance and no non-compliance alerts/events are generated.
3) Introduce deliberate deviations from the baseline (e.g. relaxing default container isolation, enabling privileged default containers, broadening default host resource access, changing a daemon setting that weakens isolation).
4) Run baseline validation again and verify that:
   - the CRS detects one or more non-compliant settings; and
   - alerts or audit log entries are generated identifying which settings deviate from the selected baseline.

5. Assessment verdict

- Pass: security baselines can be defined and used for CRS daemon configuration and default container security profiles; automatic validation against a selected baseline is available; and detected non-compliance results in alerts or audit events that identify the deviations.
- Fail: security baselines cannot be defined or do not cover both daemon config and default profiles; automatic validation cannot be performed or is ineffective; or non-compliant settings do not trigger clear alerts/audit events.

6. Assessment evidence

- [CONFIG] Example security baseline definition and corresponding CRS configuration/default profile before and after introducing deviations.
- [DESIGN] Documentation of CRS baseline management, validation logic and alert/audit integration.
- [LOG] / [OBSERVATION] Validation outputs and alert/audit records showing detection and reporting of non-compliant settings.

6.4.1.17 Assessment for Data Minimisation

6.4.1.17.1 Assessment Case AC-CRS-DM-001

1. Assessment reference
Requirement: REQ-CRS-DM-001 (Basic)

2. Assessment objective
Verify that CRS logs, telemetry, and diagnostics do not include plaintext authentication credentials, full cryptographic keys, container workload payload data, or container memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data, unless explicitly required for a documented diagnostic function and protected according to the product security documentation.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing CRS logging, telemetry, and diagnostic functions.
- Documentation identifying categories of data collected by CRS logs, telemetry, and diagnostics.
- Documentation identifying any diagnostic functions that intentionally collect sensitive data and the protections applied to such data.
- A test or demonstration configuration enabling representative CRS logs, telemetry, and diagnostic outputs.
- Tools to inspect generated logs, telemetry records, diagnostic bundles, crash reports, debug outputs, or equivalent diagnostic artefacts.

4. Assessment activities
The assessment shall at least:

1) Identify the CRS logs, telemetry, and diagnostic outputs produced by the product.
2) Generate representative logs, telemetry, and diagnostic outputs under normal and error conditions.
3) Inspect the generated outputs and verify that they do not include:

- plaintext authentication credentials;
- full cryptographic keys;
- container workload payload data;
- container memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data.

4) Where a documented diagnostic function intentionally collects sensitive data, verify that the diagnostic function is explicitly documented and that the collected data is protected according to the product security documentation.

5. Assessment verdict

- Pass: CRS logs, telemetry, and diagnostics do not include plaintext authentication credentials, full cryptographic keys, container workload payload data, or container memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data, except where explicitly required for a documented diagnostic function and protected according to the product security documentation.
- Fail: CRS logs, telemetry, or diagnostics include plaintext authentication credentials, full cryptographic keys, container workload payload data, or container memory contents or excerpts, including memory dumps, register state, pointers, or debug payloads that may expose sensitive data, and the inclusion is not explicitly required for a documented diagnostic function or is not protected according to the product security documentation.

6. Assessment evidence

- [DESIGN] Documentation of CRS logging, telemetry, and diagnostic data categories.
- [CONFIG] Configuration of CRS logging, telemetry, and diagnostic functions.
- [LOG] / [OBSERVATION] Samples of generated logs, telemetry records, diagnostic bundles, crash reports, debug outputs, or equivalent artefacts showing that prohibited sensitive data is not included.
- [DESIGN] / [CONFIG] Where sensitive diagnostic collection is intentionally supported, documentation and configuration showing the diagnostic purpose and applied protections.

6.4.1.17.2 Assessment Case AC-CRS-DM-002

1. Assessment reference
Requirement: REQ-CRS-DM-002 (Basic)

2. Assessment objective
Verify that:

1) The CRS provides administrative controls to disable or restrict collection of specific categories of log, telemetry and runtime metric data.

2) These controls allow administrators to disable or limit particular log types, telemetry streams, metric sets, or debug payload capture to align data collection with operational and regulatory requirements.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing configurable categories of logs (e.g. audit vs debug), telemetry streams, metric groups and any debug/payload capture options.
- Documentation of administrative roles/permissions required to modify data collection configuration.
- A test configuration where different log/telemetry/metric categories can be toggled and where events can be generated to exercise each category.

4. Assessment activities
The assessment shall at least:

1) Identify, from documentation and current configuration, the distinct categories controllable by administrators (e.g. specific log types, telemetry streams, metric collections, payload capture).

2) Using administrative controls, configure one or more categories (e.g. a debug log type, a telemetry stream, a metrics group, or payload capture) to be disabled or restricted.

3) Generate CRS and workload activity that would normally produce data in those categories and verify that:

- no new records are produced for disabled categories; and
- non-disabled categories continue to produce data as configured.

4) Re-enable one or more previously disabled categories and verify that data for those categories is again produced when new events occur.

5. Assessment verdict

- Pass: administrators can selectively disable or restrict specific log types, telemetry streams, metrics and any debug payload capture; disabled categories stop producing data while others continue; and re-enabling restores data collection as configured.
- Fail: data collection cannot be selectively controlled; or categories configured as disabled continue to produce data.

6. Assessment evidence

- [CONFIG] CRS data-collection configuration before and after disabling/re-enabling specific log/telemetry/metric/payload categories.
- [DESIGN] Documentation of administrative controls and category granularity for data collection, including how these support operational/regulatory alignment.
- [LOG] / [OBSERVATION] Evidence showing presence/absence of data from selected categories when enabled vs disabled, demonstrating effective restriction of collection.

6.4.1.18 Assessment for Decommissioning

6.4.1.18.1 Assessment Case AC-CRS-DECOM-001

1. Assessment reference
Requirement: REQ-CRS-DECOM-001 (Basic - Single class)

2. Assessment objective
Verify that the CRS provides a mechanism to decommission a container that renders the associated runtime state and

configuration data managed by the CRS no longer retrievable or usable through the interfaces and functions provided by the CRS.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CRS decommissioning mechanism and the runtime state and configuration data within its scope.
- A test configuration with at least one container and associated CRS-managed runtime state and configuration data accessible through CRS interfaces and functions before decommissioning.
- Access to the CRS interfaces and functions normally used to manage or retrieve that data.

4. Assessment activities
The assessment shall at least:

1) Review documentation to identify the decommissioning mechanism and the associated runtime state and configuration data managed by the CRS for the assessed container.
2) Confirm, before decommissioning, that the assessed container and its associated CRS-managed runtime state and configuration data are retrievable or usable through the interfaces and functions provided by the CRS.
3) Invoke the documented decommissioning mechanism for the container.
4) Verify, after decommissioning, that the assessed container and its associated CRS-managed runtime state and configuration data are no longer retrievable or usable through the interfaces and functions provided by the CRS.

5. Assessment verdict

- Pass: The CRS provides a decommissioning mechanism and, after its invocation, the assessed container and its associated CRS-managed runtime state and configuration data are no longer retrievable or usable through the interfaces and functions provided by the CRS.
- Fail**:** No decommissioning mechanism is provided; or, after decommissioning, the assessed container or associated CRS-managed runtime state or configuration data remain retrievable or usable through the interfaces and functions provided by the CRS.

6. Assessment evidence

- [DESIGN] Documentation of the CRS decommissioning mechanism and associated data within scope.
- [SCOPE] Identification of the assessed container and associated CRS-managed runtime state and configuration data.
- [CONFIG] Relevant CRS configuration for the assessed decommissioning procedure.
- [LOG] / [OBSERVATION] Evidence showing the accessible state before decommissioning and the non-retrievable / non-usable state after decommissioning through the interfaces and functions provided by the CRS.

6.4.2 CE

6.4.2.1 Assessment for Isolation

6.4.2.1.1 Assessment Case AC-CE-ISO-001

1. Assessment reference
Requirement: REQ-CE-ISO-001 (Basic)

2. Assessment objective
Verify that the CE provides a secure interface to configure and invoke CRS operations for the creation, execution, and

termination of containers, ensuring that the underlying CRS enforces strong logical separation consistently across all containers. Verify also that the CE prevents container workloads from accessing host administrative functions through interfaces or configuration options exposed by the CE.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CE interface for invoking CRS operations and the isolation model enforced by the underlying CRS for containers managed via the CE
- Documentation describing which CE-exposed interfaces, sockets, APIs, or configuration options could otherwise provide access to host administrative functions, and how such access is prevented for container workloads
- A test environment with the CE managing at least two containers via the CRS
- Tools to inspect CRS isolation enforcement and to attempt cross-container access and host-administrative access from CE-managed containers

4. Assessment activities
The assessment shall at least:

- Confirm that the CE interface for invoking CRS operations (creation, execution, termination) is available and functions correctly
- Verify that the CRS enforces logical separation between at least two CE-managed containers (e.g. filesystem, process, and network namespace isolation)
- Attempt to access the resources of one container from another CE-managed container and verify that the CRS rejects such access
- Attempt from a container workload to access host administrative functions through any CE-exposed interface, socket, API, or configuration option that is not explicitly intended for authorized administration, and verify that such access is prevented

5. Assessment verdict

- Pass: the CE provides a secure interface to invoke CRS operations; the CRS enforces logical separation consistently between all CE-managed containers; cross-container access attempts are rejected; and container workloads are prevented from accessing host administrative functions through CE-exposed interfaces or configuration options
- Fail: the CE interface does not enforce secure CRS invocation; or the CRS does not enforce consistent isolation between CE-managed containers; or cross-container access is possible; or container workloads can access host administrative functions through CE-exposed interfaces or configuration options

6. Assessment evidence

- [CONFIG] CE interface configuration and CRS isolation settings for CE-managed containers
- [DESIGN] Documentation of the CE-to-CRS interface architecture, isolation model, and controls preventing container workload access to host administrative functions
- [OBSERVATION] Results of cross-container access attempts and attempts by workloads to access host administrative functions through CE-exposed interfaces or options

6.4.2.1.2 Assessment Case AC-CE-ISO-002

1. Assessment reference
Requirement: REQ-CE-ISO-002 (Elevated)

2. Assessment objective
Verify that the CE enforces restrictions on CE options and configuration parameters that would weaken container isolation, including privileged execution, host namespace sharing, and unrestricted host mounts, unless explicitly enabled by an authorized administrator.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying the CE configuration options that weaken isolation (privileged mode, host namespace sharing, unrestricted host mounts) and the controls applied to those options
- A test environment with the CE in its default or hardened configuration
- Tools to attempt to launch containers with isolation-weakening options without explicit administrator authorization

4. Assessment activities
The assessment shall at least:

- Attempt to launch a container in privileged mode without explicit administrator authorization and verify that the CE rejects the request
- Attempt to launch a container with host namespace sharing (e.g. host PID, host network) without administrator authorization and verify that the CE rejects it
- Attempt to mount a sensitive host path without administrator authorization and verify that the CE rejects it
- Confirm that these options can be enabled by an authorized administrator through explicit configuration

5. Assessment verdict

- Pass: the CE restricts privileged execution, host namespace sharing, and unrestricted host mounts unless explicitly authorized by an administrator; unauthorized attempts to use these options are rejected
- Fail: the CE permits privileged execution, host namespace sharing, or unrestricted host mounts without explicit administrator authorization

6. Assessment evidence

- [CONFIG] CE policy configuration restricting isolation-weakening options
- [DESIGN] Documentation of CE isolation enforcement controls and authorized administrator override mechanism
- [LOG] / [OBSERVATION] Evidence that unauthorized privileged, namespace-sharing, and host-mount requests are rejected

6.4.2.1.3 Assessment Case AC-CE-ISO-003

1. Assessment reference
Requirement: REQ-CE-ISO-003 (Advanced)

2. Assessment objective
Verify that the CE protects its administrative interface against unauthorized access and, where the administrative interface is network-accessible, supports separation of that interface from container workload traffic and restriction of access to authorized host processes only.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CE administrative interface, its exposure (local or network-accessible), and the access controls applied
- Documentation describing how, where the administrative interface is network-accessible, it is separated from container workload traffic and restricted to authorized host processes
- A test environment with the CE administrative interface active
- Tools to attempt unauthorized access to the CE administrative interface and to inspect interface separation from container workload traffic

4. Assessment activities
The assessment shall at least:

- Confirm that the CE administrative interface is protected against unauthorized access and that unauthenticated or unauthorized access is denied
- Where the administrative interface is network-accessible, verify that it is separated from container workload traffic
- Attempt to access the administrative interface from a container workload network or workload context and verify that access is denied or restricted
- Confirm that access to the administrative interface is restricted to authorized host processes only

5. Assessment verdict

- Pass: the CE administrative interface is protected against unauthorized access; and, where network-accessible, it is separated from container workload traffic and restricted to authorized host processes only
- Fail: the CE administrative interface is accessible without authorization; or, where network-accessible, it is reachable from container workload traffic without restriction; or access is not restricted to authorized host processes

6. Assessment evidence

- [CONFIG] CE administrative interface access-control and separation configuration
- [DESIGN] Documentation of administrative interface protection and, where applicable, separation from container workload traffic and restriction to authorized host processes
- [OBSERVATION] Evidence of denied unauthorized access and effective separation from workload traffic

6.4.2.2 Assessment for Integrity Protection

6.4.2.2.1 Assessment Case AC-CE-INTEG-001

1. Assessment reference
Requirement: REQ-CE-INTEG-001 (Basic)

2. Assessment objective
Verify that the CE verifies the integrity of container images and associated metadata, including manifests, before execution, and that images failing verification are not made available to the CRS.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CE integrity verification mechanism for container images and manifests
- A test environment with the CE configured to verify image integrity
- A test image with a valid integrity check and a test image with a corrupted or mismatched integrity check

4. Assessment activities
The assessment shall at least:

- Present a container image with a valid integrity check to the CE and verify that it is accepted and made available to the CRS for execution
- Present a container image with a deliberately corrupted or mismatched integrity check and verify that the CE rejects it and does not pass it to the CRS
- Confirm that integrity verification covers both the image layers and the associated metadata/manifest

5. Assessment verdict

- Pass: container images with valid integrity checks are accepted and made available for execution; images with invalid integrity checks are rejected and not passed to the CRS
- Fail: images with invalid or mismatched integrity checks are accepted and made available to the CRS; or integrity verification does not cover both image content and metadata

6. Assessment evidence

- [CONFIG] CE image integrity verification configuration
- [DESIGN] Documentation of CE image integrity verification mechanism covering images and manifests
- [LOG] / [OBSERVATION] Evidence of acceptance of valid images and rejection of integrity-failed images

6.4.2.2.2 Assessment Case AC-CE-INTEG-002

1. Assessment reference
Requirement: REQ-CE-INTEG-002 (Elevated)

2. Assessment objective
Verify that the CE verifies the authenticity and integrity of container images or image-related artefacts using signatures validated against trusted keys, certificates, or equivalent trust anchors before use, according to the configured image trust policy.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CE image or artefact signature verification mechanism.
- Documentation identifying trusted keys, certificates, or equivalent trust anchors used for signature validation.
- Documentation describing the configured image trust policy and the point at which signature validation is performed.
- A test environment with CE signature verification configured.
- Test images or artefacts including:
  - one image or artefact signed using a trusted key, certificate, or equivalent trust anchor;
  - one image or artefact signed using an untrusted key, certificate, or trust anchor;
  - one unsigned image or artefact, where signatures are required by policy.

4. Assessment activities
The assessment shall at least:

- Review the CE signature verification mechanism and confirm that validation uses trusted keys, certificates, or equivalent trust anchors.
- Present an image or artefact signed using a trusted key, certificate, or equivalent trust anchor and verify that it is accepted according to the configured image trust policy.
- Present an image or artefact signed using an untrusted key, certificate, or trust anchor and verify that it is rejected before use.
- Present an unsigned image or artefact where signatures are required by policy and verify that it is rejected before use.

5. Assessment verdict

- Pass: the CE verifies image or artefact authenticity and integrity using signatures validated against trusted keys, certificates, or equivalent trust anchors before use according to the configured image trust policy; trusted signed images or artefacts are accepted; and untrusted or unsigned images or artefacts are rejected where required by policy.
- Fail: the CE does not validate image or artefact signatures using trusted keys, certificates, or equivalent trust anchors; or untrusted signed images or artefacts are accepted; or unsigned images or artefacts are accepted where signatures are required by policy.

6. Assessment evidence

- [DESIGN] Documentation of the CE signature verification mechanism.
- [CONFIG] Configuration of trusted keys, certificates, or equivalent trust anchors and image trust policy.
- [LOG] / [OBSERVATION] Evidence showing acceptance of trusted signed images or artefacts and rejection of untrusted or unsigned images or artefacts where required by policy.

6.4.2.2.3 Assessment Case AC-CE-INTEG-003

1. Assessment reference
Requirement: REQ-CE-INTEG-003 (Advanced)

2. Assessment objective
Verify that the CE verifies the authenticity and integrity of container images using trust anchors stored as protected trust material that is not modifiable during normal operational state, and that if verification of the integrity or authenticity of a container image using this protected trust material fails, or if the verification status of the image cannot be established, the CE prevents that image from being made available to the CRS.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing how trust anchors used for image verification are stored and protected from modification during normal operational state
- Documentation describing the CE response to verification failure or indeterminate verification status
- A test environment with protected trust anchors configured
- Tools to attempt modification of trust anchors during normal operation and to observe the response to verification failure or indeterminate verification status

4. Assessment activities
The assessment shall at least:

- Confirm that trust anchors are stored as protected trust material and that the documentation describes how modification is prevented during normal operation
- Attempt to modify the trust anchors during normal operational state (e.g. through standard administrative or workload-facing interfaces) and verify that modification is rejected or not possible
- Present an image that fails verification, or for which verification status cannot be established, and verify that the CE prevents that image from being made available to the CRS

5. Assessment verdict

- Pass: trust anchors are stored as protected material and cannot be modified through normal operational interfaces; and verification failure or indeterminate verification status prevents the image from being made available to the CRS
- Fail: trust anchors can be modified through normal operational interfaces; or verification failure or indeterminate verification status does not prevent the image from being made available to the CRS

6. Assessment evidence

- [CONFIG] Trust anchor storage and protection configuration
- [DESIGN] Documentation of protected trust material storage, modification prevention, and CE response when verification fails or cannot be established
- [LOG] / [OBSERVATION] Evidence of rejected trust-anchor modification attempts and blocked availability of the image to the CRS on verification failure or indeterminate status

6.4.2.3 Assessment for Authentication

6.4.2.3.1 Assessment Case AC-CE-AUTH-001

1. Assessment reference
Requirement: REQ-CE-AUTH-001 (Basic)

2. Assessment objective
Verify that the CE requires authentication for access to all interfaces used by orchestration components or external agents to manage containers, ensuring that only authenticated actors can execute workload lifecycle or configuration operations.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying all CE interfaces used by orchestration components or external agents for container management
- Documentation describing the authentication mechanisms applied to those interfaces
- A test environment with the CE accessible via its management interfaces
- Tools to attempt unauthenticated access to workload lifecycle or configuration operations

4. Assessment activities
The assessment shall at least:

- Enumerate CE interfaces used by orchestration components or external agents for container management, including interfaces used for workload lifecycle or configuration operations.
- Attempt to execute workload lifecycle or configuration operations without successful authentication and verify that all attempts are denied.
- Authenticate using valid credentials or an equivalent supported authentication mechanism and confirm that the permitted workload lifecycle or configuration operations succeed, subject to the applicable authorization policy.

5. Assessment verdict

- Pass: all CE interfaces used by orchestration components or external agents to manage containers require authentication; requests for workload lifecycle or configuration operations without successful authentication are denied; and successfully authenticated requests are accepted according to the applicable authorization policy.
- Fail: CE interfaces used by orchestration components or external agents to manage containers accept requests for workload lifecycle or configuration operations without successful authentication.

6. Assessment evidence

- [CONFIG] CE interface authentication configuration
- [DESIGN] Documentation of the CE interface authentication model

- [LOG] / [OBSERVATION] Evidence that requests for workload lifecycle or configuration operations without successful authentication are denied

6.4.2.3.2 Assessment Case AC-CE-AUTH-002

1. Assessment reference
Requirement: REQ-CE-AUTH-002 (Elevated)

2. Assessment objective
Verify that the CE enforces cryptographic key-based mutual authentication for communication with the orchestrator or authorized agents, and that private keys are protected from unauthorized extraction or reuse.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the mutual authentication mechanism used between the CE and the orchestrator/authorized agents
- Documentation describing how private keys used for CE authentication are protected
- A test environment with mutual authentication configured between the CE and an orchestrator
- Tools to attempt connections without valid keys and to inspect key protection

4. Assessment activities
The assessment shall at least:

- Confirm that mutual authentication is enforced between the CE and the orchestrator/agents
- Attempt to connect to the CE from a simulated orchestrator without a valid key and verify that the connection is rejected
- Confirm that the CE's private key is protected and cannot be extracted through standard operational interfaces

5. Assessment verdict

- Pass: cryptographic key-based mutual authentication is enforced; connections without valid keys are rejected; CE private keys are protected from unauthorized extraction
- Fail: connections without valid keys are accepted; or CE private keys are not protected against extraction through operational interfaces

6. Assessment evidence

- [CONFIG] Mutual authentication and key protection configuration for CE-to-orchestrator communication
- [DESIGN] Documentation of the mutual authentication architecture and private key protection mechanisms
- [LOG] / [OBSERVATION] Evidence that keyless connection attempts are rejected

6.4.2.3.3 Assessment Case AC-CE-AUTH-003

1. Assessment reference
Requirement: REQ-CE-AUTH-003 (Advanced)

2. Assessment objective
Verify that the CE enforces certificate-based mutual authentication for all management interfaces, with certificates validated against a trusted PKI including certificate validity and revocation status checks.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the PKI used for CE management interface authentication and the certificate validation checks performed (validity period, chain, revocation)

- A test environment with PKI-based mutual authentication configured on CE management interfaces
- Test certificates: valid, expired, and revoked
- Tools to inspect certificate validation behaviour

4. Assessment activities
The assessment shall at least:

- Connect using a valid certificate and verify that the connection is accepted
- Connect using an expired certificate and verify that the connection is rejected
- Connect using a revoked certificate and verify that the connection is rejected
- Connect without a certificate and verify that the connection is rejected

5. Assessment verdict

- Pass: certificate-based mutual authentication is enforced; valid certificates are accepted; expired, revoked, and absent certificates are rejected
- Fail: expired or revoked certificates are accepted; or certificate-free connections are permitted; or revocation status is not checked

6. Assessment evidence

- [CONFIG] PKI and certificate validation configuration for CE management interfaces
- [DESIGN] Documentation of PKI architecture and certificate validation logic
- [LOG] / [OBSERVATION] Test results for valid, expired, revoked, and absent certificate connection attempts

6.4.2.4 Assessment for Authorization

6.4.2.4.1 Assessment Case AC-CE-AUTHZ-001

1. Assessment reference
Requirement: REQ-CE-AUTHZ-001 (Basic)

2. Assessment objective
Verify that the CE enforces access control on all operations affecting container lifecycle or resource usage, using a default-deny policy and evaluating the authenticated identity of the caller.

3. Assessment preparation
The assessment shall have access to:

- Documentation of the CE access control model and default-deny policy
- A test environment with multiple authenticated identities with different permissions
- Tools to attempt operations as identities without the required permissions

4. Assessment activities
The assessment shall at least:

- Attempt container lifecycle operations using an authenticated identity that has not been granted permission for those operations and verify that access is denied
- Confirm that access is denied by default (i.e. access is explicitly granted)
- Confirm that authorized identities can perform permitted operations

5. Assessment verdict

- Pass: access control is enforced on container lifecycle and resource operations; a default-deny policy is applied; the authenticated identity of the caller is evaluated for each operation
- Fail: container lifecycle or resource operations can be performed by unauthorized identities; or a default-permit policy is applied

6. Assessment evidence

- [CONFIG] CE access control configuration and default-deny policy settings
- [DESIGN] Documentation of CE access control model
- [LOG] / [OBSERVATION] Evidence that unauthorized identity access attempts are denied

6.4.2.4.2 Assessment Case AC-CE-AUTHZ-002

1. Assessment reference
Requirement: REQ-CE-AUTHZ-002 (Elevated)

2. Assessment objective
Verify that the CE enforces granular authorization for container lifecycle and resource-usage control operations, binding permitted actions to the authenticated identity of the caller and enforcing least-privilege permissions by default. The assessment shall not require a specific authorization model, provided that the implemented model supports granular permissions and least-privilege enforcement. The implemented authorization model may be role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the implemented CE authorization model, including whether it is role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.
- Documentation identifying CE operations that affect container lifecycle or resource usage.
- Documentation identifying authenticated caller identities, permissions, policy rules, or equivalent authorization bindings.
- A test configuration with at least two authenticated caller identities having different permissions for representative CE control operations.
- Tools or interfaces to attempt permitted and non-permitted CE control operations using those authenticated identities.

4. Assessment activities
The assessment shall at least:

- Identify the CE operations covered by REQ-CE-AUTHZ-002, including representative operations that affect container lifecycle or resource usage.
- Review the implemented authorization model and confirm that permitted actions are bound to authenticated caller identities.
- Verify that least-privilege permissions are enforced by default.
- Attempt a permitted CE control operation using an authenticated identity authorized for that action and verify that the action succeeds.
- Attempt the same CE control operation using an authenticated identity not authorized for that action and verify that the action is denied.
- Verify that the evidence demonstrates granular authorization without requiring a specific authorization model.

5. Assessment verdict

- Pass: the CE enforces granular authorization for container lifecycle and resource-usage control operations, binds permitted actions to authenticated caller identities, and enforces least-privilege permissions by default, regardless of whether the implemented model is role-based, attribute-based, policy-based, capability-based, or equivalent.
- Fail: the CE does not enforce granular authorization for container lifecycle or resource-usage control operations; or permitted actions are not bound to authenticated caller identities; or least-privilege permissions are not enforced by default; or an authenticated caller can perform an action for which it is not authorized.

6. Assessment evidence

- [DESIGN] Documentation of the implemented CE authorization model and its binding of permissions to authenticated caller identities.
- [CONFIG] Authorization policy, role, attribute, rule, capability, or equivalent configuration showing permitted and non-permitted CE control operations.
- [LOG] / [OBSERVATION] Evidence of permitted CE control operations succeeding for authorized identities and non-permitted CE control operations being denied for unauthorized identities.

6.4.2.4.3 Assessment Case AC-CE-AUTHZ-003

1. Assessment reference
Requirement: REQ-CE-AUTHZ-003 (Advanced)

2. Assessment objective
Verify that the CE enforces fine-grained, policy-driven, or context-aware authorization for container lifecycle and resource-usage control operations, including contextual policy evaluation where applicable to the corresponding requirement and declared product configuration. The assessment shall not require ABAC specifically, provided that the implemented model enforces the required fine-grained, policy-driven, or context-aware authorization properties.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CE fine-grained, policy-driven, or context-aware authorization model.
- Documentation identifying protected CE operations affecting container lifecycle or resource usage.
- Documentation identifying contextual policy inputs, policy rules, authorization policies, or equivalent mechanisms supported by the CE, where applicable.
- A test configuration with representative authenticated caller identities, permissions, contextual attributes, or policy inputs, as applicable to the declared product configuration.

4. Assessment activities
The assessment shall at least:

- Review the documented fine-grained, policy-driven, or context-aware CE authorization model.
- Verify that the CE enforces the applicable authorization policy for representative container lifecycle and resource-usage operations.
- Where context-aware enforcement is supported or applicable, verify that authorization decisions take the documented contextual inputs into account.
- Attempt a protected CE operation under an authorized policy condition and verify that the operation succeeds.
- Attempt the same protected CE operation under an unauthorized policy condition and verify that the operation is denied.

5. Assessment verdict

- Pass: the CE enforces fine-grained, policy-driven, or context-aware authorization for container lifecycle and resource-usage control operations, including contextual policy evaluation where applicable to the requirement and declared product configuration.
- Fail: the CE does not enforce fine-grained, policy-driven, or context-aware authorization where required; or contextual policy evaluation is claimed but not implemented or not effective in the declared product configuration; or an unauthorized caller or unauthorized policy condition allows execution of a protected CE operation.

6. Assessment evidence

- [DESIGN] Documentation of the fine-grained, policy-driven, or context-aware CE authorization model.
- [CONFIG] Authorization policy, contextual policy, or equivalent configuration, as applicable.
- [LOG] / [OBSERVATION] Evidence that CE authorization decisions are enforced according to the documented policy and declared product configuration.

6.4.2.5 Assessment for Confidentiality and Data Protection

6.4.2.5.1 Assessment Case AC-CE-CONF-001

1. Assessment reference
Requirement: REQ-CE-CONF-001 (Basic)

2. Assessment objective
Verify that the CE prevents unauthorized access to container images, configuration data, and secrets stored or managed by the CE, by enforcing CE-controlled access boundaries.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying where container images, configuration data, and secrets are stored or managed by the CE and the access controls applied
- A test environment with container images, configuration data, and secrets present in CE-managed storage
- Tools to attempt unauthorized access to CE-managed images, configuration data, and secrets

4. Assessment activities
The assessment shall at least:

- Identify storage locations for CE-managed images, configuration data, and secrets
- Attempt to access CE-managed images or secrets without CE-controlled authorization (e.g. as an unauthenticated user or as a container process without granted access) and verify that access is denied
- Confirm that CE-controlled access boundaries prevent exposure of these resources outside the CE's access control scope

5. Assessment verdict

- Pass: unauthorized access to CE-managed images, configuration data, and secrets is prevented by CE-controlled access boundaries
- Fail: CE-managed images, configuration data, or secrets are accessible without CE-controlled authorization

6. Assessment evidence

- [CONFIG] CE access control configuration for image, configuration, and secrets storage
- [DESIGN] Documentation of CE-controlled access boundaries

- [OBSERVATION] Evidence that unauthorized access attempts to CE-managed resources are denied

6.4.2.5.2 Assessment Case AC-CE-CONF-002

1. Assessment reference
Requirement: REQ-CE-CONF-002 (Elevated)

2. Assessment objective
Verify that the CE supports encryption at rest for container images, configuration data, and secrets under its control, and ensures that CE administrative interfaces and communications with external services are protected using encrypted channels.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing support for encryption at rest for CE-managed images, configuration data, and secrets
- Documentation describing protection of CE administrative interfaces and communications with external services using encrypted channels
- A test environment with the relevant encryption-at-rest and encrypted-channel functions configured
- Tools to inspect protected storage and to capture or inspect CE communication traffic

4. Assessment activities
The assessment shall at least:

- Confirm that the CE supports encryption at rest for container images, configuration data, and secrets under its control and, where this capability is configured in the assessed environment, verify that the relevant data is stored in protected form
- Verify that CE administrative interface communications are protected using encrypted channels
- Capture or inspect CE communications with external services and confirm that, where such communications are in scope, they are protected using encrypted channels

5. Assessment verdict

- Pass: the CE supports encryption at rest for container images, configuration data, and secrets under its control; and CE administrative interfaces and communications with external services are protected using encrypted channels in the assessed configuration
- Fail: the CE does not support encryption at rest for container images, configuration data, and secrets under its control; or CE administrative interfaces or communications with external services are not protected using encrypted channels in the assessed configuration

6. Assessment evidence

- [CONFIG] Encryption-at-rest and encrypted-channel configuration
- [DESIGN] Documentation of CE support for encryption of managed data at rest and protection of administrative interfaces and external communications
- [OBSERVATION] Evidence of protected storage where configured and protected communication channels for CE administrative and external-service communications

6.4.2.5.3 Assessment Case AC-CE-CONF-003

1. Assessment reference
Requirement: REQ-CE-CONF-003 (Advanced)

2. Assessment objective
Verify that, where the underlying platform exposes protected execution contexts that provide confidentiality of

workload data in use against privileged software, the CE provides configuration and API interfaces to enable workload execution in such protected contexts and, where the user opts to use these capabilities, enforces placement and associated security policies such that declared confidential workloads run only in such protected contexts. Verify also that the CE supports secure provisioning of secrets and key material into the protected execution context such that they are not exposed outside the protected provisioning path to the host operating system, other workloads, or CE components not authorized to handle those materials.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing CE support for platform-provided confidential execution environments and the CE configuration or API interfaces for enabling workload execution in those environments.
- Documentation of the placement rules, isolation constraints, and associated security policies enforced by the CE for declared confidential workloads.
- Documentation of the mechanism used to securely provision secrets and key material into the protected execution context.
- Documentation identifying CE components authorized to handle secrets or key material during protected provisioning, and CE components not authorized to handle those materials.
- A test environment with a platform supporting confidential execution and the CE configured to use it.
- Tools to schedule a workload in a confidential execution environment, verify placement and isolation constraints, and assess access to provisioned secrets and key material through host operating system interfaces, other workloads, and CE components or interfaces not authorized to handle those materials.

4. Assessment activities
The assessment shall at least:

- Review the CE interfaces and configuration parameters used to enable workload execution in protected execution contexts.
- Configure a workload as a declared confidential workload and verify that the CE places it only in a protected execution context according to the documented placement and associated security policies.
- Attempt to schedule or run the declared confidential workload outside the protected execution context and verify that this is prevented or rejected according to the documented policy.
- Provision secrets or key material into the protected execution context using the documented mechanism.
- Attempt to access the provisioned secrets or key material through host operating system interfaces, other workloads, and CE components or interfaces not authorized to handle those materials and verify that such access is prevented or that the material is not exposed outside the protected provisioning path.

5. Assessment verdict

- Pass: the CE provides interfaces to enable workload execution in platform-supported confidential execution environments; declared confidential workloads are placed only in such protected contexts with the documented placement and associated security policies enforced; and secrets and key material can be securely provisioned into the protected execution context without being exposed outside the protected provisioning path to the host operating system, other workloads, or CE components not authorized to handle those materials.
- Fail: the CE does not provide interfaces for confidential execution; or declared confidential workloads are not constrained to protected execution contexts as required; or placement and associated security policies are not enforced; or provisioned secrets or key material are exposed outside the protected provisioning path to the host operating system, other workloads, or CE components not authorized to handle those materials.

6. Assessment evidence

- [CONFIG] CE confidential-execution scheduling, placement, and associated security-policy configuration.
- [DESIGN] Documentation of CE support for confidential execution environments and placement enforcement.

- [DESIGN] Documentation of the protected provisioning path for secrets and key material, including CE components authorized to handle those materials.
- [PLT-CAP] Platform capability evidence for confidential execution environment support.
- [OBSERVATION] Evidence of workload execution in the protected context and enforcement of placement and associated policies.
- [OBSERVATION] Evidence that attempts to access provisioned secrets or key material through host operating system interfaces, other workloads, and CE components or interfaces not authorized to handle those materials are prevented or do not expose the material outside the protected provisioning path.

6.4.2.6 Assessment for Secure Update

6.4.2.6.1 Assessment Case AC-CE-UPD-001

1. Assessment reference
Requirement: REQ-CE-UPD-001 (Basic)

2. Assessment objective
Verify that:

- The CE is delivered in a form that supports authenticity and integrity verification of CE updates before installation or activation.
- The CE supports authenticity and integrity verification of CE updates using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Where the CE includes the mechanism that verifies, installs, or activates its updates, this mechanism prevents installation or activation of an update when authenticity or integrity verification fails and records a security-relevant event.
- Where update delivery, verification, installation, or activation is performed by a Host OS update service, package manager, platform update service, appliance update mechanism, CO, orchestration-based update pipeline, or other Operational Environment mechanism outside the declared CE product or outside the responsibility of the manufacturer, the product security documentation identifies that mechanism, the expected security properties, and the assumptions required for secure update of the CE.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying the CE components delivered as part of the declared CE product.
- Documentation describing the update artefacts applicable to those CE components.
- Documentation describing how authenticity and integrity verification of CE updates is supported before installation or activation, including the verification metadata used, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Documentation identifying whether update delivery, verification, installation, or activation is performed by the CE, by another component under the responsibility of the manufacturer, or by an Operational Environment mechanism.
- Where the CE includes the mechanism that verifies, installs, or activates updates, a test configuration allowing use of a valid update artefact and an update artefact with invalid, missing, revoked, or intentionally modified authenticity or integrity protection, or an equivalent test condition.
- Where update delivery, verification, installation, or activation is performed by an Operational Environment mechanism, product security documentation identifying the expected update mechanism, expected security properties, and assumptions required for secure update of the CE.

4. Assessment activities
The assessment shall at least:

- Identify the CE components and update artefacts covered by REQ-CE-UPD-001.
- Check that the CE is delivered in a form that supports authenticity and integrity verification of CE updates before installation or activation.
- Check that the update artefacts support authenticity and integrity verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Check whether update delivery, verification, installation, or activation is performed by the CE, by another component under the responsibility of the manufacturer, or by an Operational Environment mechanism.
- Where the CE includes the mechanism that verifies, installs, or activates updates, attempt to install or activate an update whose authenticity or integrity verification fails and verify that installation or activation is prevented and that a security-relevant event is recorded.
- Where update delivery, verification, installation, or activation is performed outside the declared CE product or outside the responsibility of the manufacturer, verify that the product security documentation identifies the Operational Environment dependency, the expected security properties, and the assumptions required for secure update of the CE.

5. Assessment verdict

- Pass: the CE is delivered in a form that supports authenticity and integrity verification of CE updates before installation or activation; update artefacts support verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata; and, where the CE includes the mechanism that verifies, installs, or activates updates, failed authenticity or integrity verification prevents installation or activation and is recorded as a security-relevant event. Where update delivery, verification, installation, or activation is performed outside the declared CE product or outside the responsibility of the manufacturer, the Operational Environment dependency, expected security properties, and assumptions are documented.
- Fail: the CE is not delivered in a form that supports authenticity and integrity verification of CE updates before installation or activation; or the update artefacts do not support the required verification metadata; or, where the CE includes the mechanism that verifies, installs, or activates updates, an update whose authenticity or integrity verification fails can be installed or activated, or the failure is not recorded as a security-relevant event; or the relevant Operational Environment dependency is not documented where update delivery, verification, installation, or activation is performed outside the declared CE product or outside the responsibility of the manufacturer.

6. Assessment evidence

- [DESIGN] Documentation of the CE update model, including update artefacts and the component or mechanism responsible for update delivery, verification, installation, or activation.
- [DESIGN] Documentation of authenticity and integrity verification metadata used for CE updates, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- [CONFIG] Configuration of the CE update mechanism, where the CE includes the mechanism that verifies, installs, or activates updates.
- [LOG] / [OBSERVATION] Where the CE includes the mechanism that verifies, installs, or activates updates, evidence that a valid update is accepted and that an update with failed authenticity or integrity verification is prevented from installation or activation and recorded as a security-relevant event.
- [GUIDANCE] / [OE] Where update delivery, verification, installation, or activation is performed outside the declared CE product or outside the responsibility of the manufacturer, product security documentation identifying the Operational Environment dependency, expected security properties, and assumptions required for secure update of the CE.

6.4.2.6.2 Assessment Case AC-CE-UPD-002

1. Assessment reference

Requirement: REQ-CE-UPD-002 (Elevated)

2. Assessment objective

Verify that

- The CE supports rollback protection for CE updates where rollback protection is provided by the CE or by an update mechanism under the responsibility of the manufacturer.
- Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed CE updates is prevented.
- Authorized rollback to a previously trusted CE version is permitted only when:
  a) appropriate privileges or an equivalent authorized control function; and
  b) the rollback operation is recorded as a security-relevant event appropriate privileges or an equivalent authorized control function; and
  c) the rollback operation is recorded as a security-relevant event.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying whether rollback protection for CE updates is provided by the CE or by an update mechanism under the responsibility of the manufacturer.
- Documentation describing the rollback protection mechanism, where rollback protection is provided.
- Documentation describing procedures for authorized rollback to a previously trusted CE version, where authorized rollback is supported.
- Documentation identifying the administrator privileges or equivalent authorized control function required to authorize rollback, where authorized rollback is supported.
- Documentation describing how rollback operations are recorded as security-relevant events, where rollback is supported.
- A test or demonstration configuration allowing an outdated, revoked, or replayed CE update to be attempted, where rollback protection is provided by the CE or by an update mechanism under the responsibility of the manufacturer.
- A test or demonstration configuration allowing authorized rollback to a previously trusted CE version, where authorized rollback is supported.

4. Assessment activities
The assessment shall at least:

- Check whether rollback protection for CE updates is provided by the CE or by an update mechanism under the responsibility of the manufacturer.
- Where rollback protection is not provided by the CE or by an update mechanism under the responsibility of the manufacturer, verify that the product security documentation identifies the update and rollback assumptions applicable to the declared product configuration.
- Where rollback protection is provided, attempt to install an outdated, revoked, or replayed CE update outside the documented authorized rollback procedure and verify that installation is prevented.
- Where authorized rollback is supported, perform an authorized rollback using the documented procedure and verify that:
  a) the rollback image or package is verified for integrity and authenticity;

b) the rollback action requires explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; and

c) the rollback operation is recorded as a security-relevant event.

5. Assessment verdict

- Pass: where rollback protection is provided by the CE or by an update mechanism under the responsibility of the manufacturer, unauthorized installation of outdated, revoked, or replayed CE updates is prevented; and authorized rollback to a previously trusted CE version is permitted only following integrity and authenticity verification of the rollback image or package, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function, and recording of the rollback operation as a security-relevant event. Where rollback protection is not provided by the CE or by an update mechanism under the responsibility of the manufacturer, the update and rollback assumptions applicable to the declared product configuration are documented.

- Fail: where rollback protection is provided by the CE or by an update mechanism under the responsibility of the manufacturer, outdated, revoked, or replayed CE updates can be installed without authorization; or authorized rollback can be performed without integrity and authenticity verification; or rollback can be performed without explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; or the rollback operation is not recorded as a security-relevant event.

6. Assessment evidence

- [DESIGN] Documentation identifying whether rollback protection is provided by the CE or by an update mechanism under the responsibility of the manufacturer.

- [DESIGN] Documentation of rollback protection mechanisms and authorized rollback procedures, where rollback protection or authorized rollback is supported.

- [CONFIG] Configuration or policy information showing rollback authorization controls, where authorized rollback is supported.

- [LOG] / [OBSERVATION] Where rollback protection is provided, evidence that outdated, revoked, or replayed updates are prevented outside the authorized rollback procedure.

- [LOG] / [OBSERVATION] Where authorized rollback is supported, evidence that rollback is performed only after integrity and authenticity verification, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function, and recording as a security-relevant event.

- [GUIDANCE] / [OE] Where rollback protection is not provided by the CE or by an update mechanism under the responsibility of the manufacturer, documentation identifying update and rollback assumptions applicable to the declared product configuration.

6.4.2.7 Assessment for Decommissioning and Secure Transfer

6.4.2.7.1 Assessment Case AC-CE-DECOM-001

1. Assessment reference
Requirement: REQ-CE-DECOM-001 (Basic - Single class)

2. Assessment objective
Verify that the CE provides a mechanism to remove container images, cached image layers, and associated configuration data managed by the CE that renders such data no longer retrievable or usable through the interfaces and functions provided by the CE.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CE removal mechanism and the container images, cached image layers, and associated configuration data within its scope.

- A test configuration with at least one removable image, cached image layer, and associated CE-managed configuration data accessible through CE interfaces and functions before removal.

- Access to the CE interfaces and functions normally used to manage or retrieve that data.

4. Assessment activities
The assessment shall at least:

1) Review documentation to identify the removal mechanism and the container images, cached image layers, and associated configuration data managed by the CE for the assessment.

2) Confirm, before removal, that the assessed data is retrievable or usable through the interfaces and functions provided by the CE.

3) Invoke the documented removal mechanism.

4) Verify, after removal, that the assessed data is no longer retrievable or usable through the interfaces and functions provided by the CE.

5. Assessment verdict

- Pass: The CE provides a removal mechanism and, after its invocation, the assessed container images, cached image layers, and associated configuration data are no longer retrievable or usable through the interfaces and functions provided by the CE.

- Fail: No removal mechanism is provided; or, after removal, the assessed data remains retrievable or usable through the interfaces and functions provided by the CE.

6. Assessment evidence

- [DESIGN] Documentation of the CE removal mechanism and associated data within scope.

- [SCOPE] Identification of the assessed container images, cached image layers, and associated configuration data.

- [CONFIG] Relevant CE configuration for the assessed removal procedure.

- [LOG] / [OBSERVATION] Evidence showing the accessible state before removal and the non-retrievable / non-usable state after removal through the interfaces and functions provided by the CE.

6.4.2.7.2 Assessment Case AC-CE-DECOM-002

1. Assessment reference
Requirement: REQ-CE-DECOM-002 (Basic - Single class)

2. Assessment objective
Verify that, where the CE supports transfer of container image metadata, configuration, or associated management data to another product or system, the CE protects the confidentiality and integrity of the transferred data through the interfaces and functions provided by the CE.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the supported transfer mechanism and the types of container image metadata, configuration, or associated management data that can be transferred.

- Documentation describing the security protections applied to the transfer.

- A test configuration in which supported data can be transferred from the CE to another product or system.

- Access to the interfaces and functions provided by the CE for the transfer operation.

4. Assessment activities
The assessment shall at least:

1) Review documentation to identify the supported transfer mechanism and the security protections applied to the transferred data.

2) Invoke the documented transfer mechanism for supported container image metadata, configuration, or associated management data.

3) Verify that the transfer is performed through the interfaces and functions provided by the CE with protection of the confidentiality and integrity of the transferred data.

4) Where feasible, verify that transfer without the documented protection is not permitted through the interfaces and functions provided by the CE.

5. Assessment verdict

- Pass: Where the CE supports transfer, the documented transfer mechanism protects the confidentiality and integrity of the transferred data through the interfaces and functions provided by the CE.
- Fail: A supported transfer mechanism does not protect the confidentiality or integrity of the transferred data; or transfer can be performed through the interfaces and functions provided by the CE without the documented protection.

6. Assessment evidence

- [DESIGN] Documentation of the transfer mechanism and associated security protections.
- [SCOPE] Identification of the transferred data types assessed.
- [CONFIG] Relevant CE configuration for the assessed transfer procedure.
- [LOG] / [OBSERVATION] Evidence showing the protected transfer of the assessed data through the interfaces and functions provided by the CE.

6.4.3 CO

6.4.3.1 Assessment for Isolation

6.4.3.1.1 Assessment Case AC-CO-ISO-001

1. Assessment reference
Requirement: REQ-CO-ISO-001 (Basic)

2. Assessment objective
Verify that the CO enforces strong logical network isolation between workloads scheduled in different namespaces or tenancy domains, using network isolation policies or equivalent orchestration mechanisms.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the network isolation policies or equivalent mechanisms used by the CO to enforce isolation between workloads in different namespaces or tenancy domains
- A test environment with workloads deployed in at least two different namespaces or tenancy domains
- Tools to attempt cross-namespace/cross-domain network communication and to inspect policy configuration

4. Assessment activities
The assessment shall at least:

- Confirm that network isolation policies or equivalent mechanisms are configured to restrict network traffic between workloads in different namespaces or tenancy domains
- Attempt to send network traffic from a workload in namespace A to a workload in namespace B without an explicit allow policy and verify that the traffic is blocked
- Confirm that explicitly permitted cross-namespace communication is allowed when a policy grants it

5. Assessment verdict

- Pass: network isolation between workloads in different namespaces/tenancy domains is enforced; unauthorized cross-namespace traffic is blocked; explicitly permitted communication is allowed
- Fail: network traffic between workloads in different namespaces or tenancy domains is not restricted; or isolation policies are not enforced

6. Assessment evidence

- [CONFIG] Network isolation policy configuration for namespaces/tenancy domains
- [DESIGN] Documentation of CO network isolation model
- [OBSERVATION] Evidence of blocked cross-namespace traffic and permitted authorized communication

6.4.3.1.2 Assessment Case AC-CO-ISO-002

1. Assessment reference
Requirement: REQ-CO-ISO-002 (Elevated)

2. Assessment objective
Verify that the CO logically isolates its management and orchestration interfaces (API server) from the container data plane to prevent unauthorized pivot attempts from compromised workloads.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing how the CO management/API interfaces are logically isolated from the container data plane
- A test environment with workloads running and CO management interfaces active
- Tools to attempt to reach the CO API server from a container workload network

4. Assessment activities
The assessment shall at least:

- Confirm that the CO management/API interfaces are logically separated from the container data plane (e.g. via network policy, dedicated management network, or firewall rules)
- Attempt to reach the CO API server from within a container workload and verify that direct access is denied or restricted
- Confirm that service accounts used by workloads have only the minimum required access to CO APIs

5. Assessment verdict

- Pass: the CO management/API interfaces are logically isolated from the container data plane; unauthorized access from container workloads to the API server is denied or restricted
- Fail: the CO management/API interfaces are reachable from container workloads without restriction; or pivot attempts from workloads to the control plane are not prevented

6. Assessment evidence

- [CONFIG] Network policy or firewall configuration isolating CO management interfaces from the data plane
- [DESIGN] Documentation of CO management/data plane isolation architecture
- [OBSERVATION] Evidence that CO API server is not directly accessible from container workloads

6.4.3.1.3 Assessment Case AC-CO-ISO-003

1. Assessment reference
Requirement: REQ-CO-ISO-003 (Advanced)

2. Assessment objective
Verify that the CO enforces cryptographic protection on control plane communications and ensures that control plane interfaces cannot be accessed from workload or data plane networks.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing cryptographic protection of CO control plane communications
- Documentation describing the network controls preventing control plane interface access from workload/data plane networks
- A test environment with cryptographic protection and network controls configured
- Tools to capture control plane traffic and to attempt access from workload networks

4. Assessment activities
The assessment shall at least:

- Capture CO control plane communications and verify that they are encrypted using appropriate cryptographic mechanisms
- Attempt to access CO control plane interfaces from a workload/data plane network and verify that access is denied
- Confirm that the cryptographic protection covers both confidentiality and integrity of control plane communications

5. Assessment verdict

- Pass: CO control plane communications are cryptographically protected; control plane interfaces cannot be accessed from workload/data plane networks
- Fail: control plane communications are not cryptographically protected; or control plane interfaces are accessible from workload/data plane networks

6. Assessment evidence

- [CONFIG] Cryptographic protection and network control configuration for CO control plane
- [DESIGN] Documentation of control plane communication security and network isolation
- [OBSERVATION] Encrypted control plane traffic capture and evidence of denied workload-network access to control plane interfaces

6.4.3.2 Assessment for Integrity Protection

6.4.3.2.1 Assessment Case AC-CO-INTEG-001

1. Assessment reference
Requirement: REQ-CO-INTEG-001 (Basic)

2. Assessment objective
Verify that the CO protects the integrity of its configuration and policy state, ensuring that only authorized components and roles can modify control plane configuration, workload specifications, or cluster policies.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the access controls protecting CO configuration and policy state

- A test environment with CO configuration and cluster policies in place
- Tools to attempt unauthorized modification of CO configuration and policies

4. Assessment activities
The assessment shall at least:

- Attempt to modify CO control plane configuration, a workload specification, or a cluster policy using an account without the required authorization and verify that the modification is denied
- Confirm that authorized modifications succeed and are reflected in the CO state
- Review audit logs to confirm that unauthorized modification attempts are recorded

5. Assessment verdict

- Pass: CO configuration, workload specifications, and cluster policies can only be modified by authorized components and roles; unauthorized modification attempts are denied and logged
- Fail: CO configuration or policies can be modified by unauthorized accounts or components

6. Assessment evidence

- [CONFIG] Access control configuration protecting CO configuration and policy state
- [DESIGN] Documentation of CO configuration integrity controls
- [LOG] / [OBSERVATION] Audit log entries for authorized and denied configuration modification attempts

6.4.3.2.2 Assessment Case AC-CO-INTEG-002

1. Assessment reference
Requirement: REQ-CO-INTEG-002 (Elevated)

2. Assessment objective
Verify that the CO verifies the integrity of cluster configuration artefacts, deployment specifications, admission policies, policy bundles, or equivalent orchestrator-controlled artefacts using cryptographic digests, digital signatures, or equivalent trusted reference values before applying them to cluster state.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CO integrity verification mechanism for cluster configuration artefacts, deployment specifications, admission policies, policy bundles, or equivalent orchestrator-controlled artefacts.
- Documentation identifying the cryptographic digests, digital signatures, or equivalent trusted reference values used for verification.
- A test environment with representative cluster configuration artefacts or equivalent orchestrator-controlled artefacts.
- Test artefacts including:
  - one valid artefact with a valid cryptographic digest, digital signature, or equivalent trusted reference value;
  - one modified artefact with an invalid or mismatched digest, signature, or trusted reference value;
  - one artefact without required integrity metadata, where applicable to the configured policy.

4. Assessment activities
The assessment shall at least:

- Review the CO integrity verification mechanism and the trusted reference values used for cluster configuration artefacts or equivalent orchestrator-controlled artefacts.

- Present a valid artefact with a valid cryptographic digest, digital signature, or equivalent trusted reference value and verify that it is accepted.
- Present a modified artefact with an invalid or mismatched digest, signature, or trusted reference value and verify that it is rejected before being applied to cluster state.
- Where integrity metadata is required by policy, present an artefact without the required integrity metadata and verify that it is rejected before being applied to cluster state.

5. Assessment verdict

- Pass: the CO verifies cluster configuration artefacts or equivalent orchestrator-controlled artefacts using cryptographic digests, digital signatures, or equivalent trusted reference values before applying them to cluster state; valid artefacts are accepted; and artefacts with invalid, mismatched, or missing required integrity metadata are rejected according to policy.
- Fail: the CO does not verify cluster configuration artefacts or equivalent orchestrator-controlled artefacts using cryptographic digests, digital signatures, or equivalent trusted reference values; or artefacts with invalid, mismatched, or missing required integrity metadata are applied to cluster state contrary to the configured policy.

6. Assessment evidence

- [DESIGN] Documentation of the CO artefact integrity verification mechanism.
- [CONFIG] Configuration of cryptographic digests, digital signatures, or equivalent trusted reference values used for artefact verification.
- [LOG] / [OBSERVATION] Evidence showing acceptance of valid artefacts and rejection of invalid, mismatched, or missing-integrity artefacts according to policy.

6.4.3.2.3 Assessment Case AC-CO-INTEG-003

1. Assessment reference
Requirement: REQ-CO-INTEG-003 (Advanced)

2. Assessment objective
Verify that the CO enforces integrity of workload deployments at admission time by applying admission control policies and preventing deployment of workloads that violate integrity, authorization, or configuration constraints.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CO admission control mechanism and the types of policies enforced at admission (integrity, authorization, configuration constraints)
- A test environment with admission control policies configured
- Workload deployment requests: one compliant with all policies and one violating at least one policy

4. Assessment activities
The assessment shall at least:

- Submit a compliant workload deployment request and verify that it is accepted by the admission control mechanism
- Submit a workload deployment request that violates an integrity, authorization, or configuration constraint and verify that it is rejected at admission
- Confirm that the admission control mechanism is active and applied to all workload deployments

5. Assessment verdict

- Pass: compliant workload deployments are accepted; deployments violating integrity, authorization, or configuration constraints are rejected at admission; admission control is applied to all deployments

- Fail: deployments violating policy constraints are accepted; or admission control is not consistently applied to all deployments

6. Assessment evidence

- [CONFIG] Admission control policy configuration
- [DESIGN] Documentation of CO admission control architecture and policy types
- [LOG] / [OBSERVATION] Admission control decision records for compliant and policy-violating deployment requests

6.4.3.3 Assessment for Authentication

6.4.3.3.1 Assessment Case AC-CO-AUTH-001

1. Assessment reference
Requirement: REQ-CO-AUTH-001 (Basic)

2. Assessment objective
Verify that the CO enforces authenticated access for all administrative and user interactions with control plane APIs and management interfaces, and that password strength and lockout policies are applied where password-based authentication is used.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying CO control plane APIs and management interfaces and the authentication mechanisms applied
- Documentation of password strength and lockout policies, where applicable
- A test environment with the CO control plane APIs and management interfaces active
- Tools to attempt unauthenticated access and to test password policy enforcement

4. Assessment activities
The assessment shall at least:

- Attempt to access CO control plane APIs and management interfaces without authentication and verify that access is denied
- Where password-based authentication is used, confirm that password strength policies are enforced
- Where password-based authentication is used, confirm that a lockout policy is applied after repeated failed attempts

5. Assessment verdict

- Pass: CO control plane APIs and management interfaces require authentication; unauthenticated access is denied; password strength and lockout policies are enforced where applicable
- Fail: CO control plane APIs or management interfaces accept unauthenticated access; or password strength/lockout policies are not enforced where password authentication is used

6. Assessment evidence

- [CONFIG] Authentication configuration for CO control plane APIs and management interfaces
- [DESIGN] Documentation of CO authentication model and, where applicable, password policies
- [LOG] / [OBSERVATION] Evidence that unauthenticated access attempts are denied

6.4.3.3.2 Assessment Case AC-CO-AUTH-002

1. Assessment reference
Requirement: REQ-CO-AUTH-002 (Elevated)

2. Assessment objective
Verify that the CO enforces cryptographically verifiable authentication mechanisms for service accounts and administrative identities and requires mutual authentication for internal control plane communication.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing cryptographically verifiable authentication for CO service accounts and administrative identities
- Documentation describing mutual authentication for internal CO control plane communication
- A test environment with cryptographic authentication and mutual authentication configured
- Tools to inspect authentication mechanism usage and to attempt connections without valid credentials

4. Assessment activities
The assessment shall at least:

- Confirm that service account and administrative identity authentication uses a cryptographically verifiable mechanism (e.g. token with cryptographic signature, certificate)
- Confirm that internal CO control plane communication requires mutual authentication
- Attempt to connect to the CO control plane without valid credentials and verify that access is denied

5. Assessment verdict

- Pass: cryptographically verifiable authentication is enforced for service accounts and administrative identities; mutual authentication is required for internal control plane communication
- Fail: non-cryptographic authentication is used; or internal control plane communication does not require mutual authentication

6. Assessment evidence

- [CONFIG] Authentication configuration for CO service accounts and internal control plane communication
- [DESIGN] Documentation of cryptographic authentication and mutual authentication architecture
- [LOG] / [OBSERVATION] Evidence that access without valid cryptographic credentials is denied

6.4.3.3.3 Assessment Case AC-CO-AUTH-003

1. Assessment reference
Requirement: REQ-CO-AUTH-003 (Advanced)

2. Assessment objective
Verify that the CO enforces certificate-based authentication across all control plane and administrative interfaces, and supports integration with a trusted PKI for issuing and managing certificates.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing PKI-based certificate authentication across CO control plane and administrative interfaces
- Documentation describing PKI integration for certificate issuance and management
- A test environment with PKI-based authentication configured

- Test certificates: valid, expired, revoked

4. Assessment activities
The assessment shall at least:

- Confirm that certificate-based authentication is required across all CO control plane and administrative interfaces
- Connect using a valid PKI-issued certificate and verify that access is granted
- Connect using an expired or revoked certificate and verify that access is denied
- Confirm that the CO integrates with the configured trusted PKI for certificate operations

5. Assessment verdict

- Pass: certificate-based authentication is enforced across CO control plane and administrative interfaces; valid certificates are accepted; expired/revoked certificates are rejected; PKI integration is operational
- Fail: certificate-based authentication is not enforced; expired or revoked certificates are accepted; or PKI integration does not function correctly

6. Assessment evidence

- [CONFIG] PKI and certificate authentication configuration for CO control plane and administrative interfaces
- [DESIGN] Documentation of CO PKI integration architecture
- [LOG] / [OBSERVATION] Test results for valid, expired, and revoked certificate authentication attempts

6.4.3.4 Assessment for Authorization

6.4.3.4.1 Assessment Case AC-CO-AUTHZ-001

1. Assessment reference
Requirement: REQ-CO-AUTHZ-001 (Basic)

2. Assessment objective
Verify that the CO enforces a default-deny authorization policy for all control plane actions and resource access, restricting actions to authenticated entities.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the default-deny authorization policy for CO control plane actions and resource access
- A test environment with authenticated accounts having varying authorization levels
- Tools to attempt control plane actions and resource access without explicit authorization

4. Assessment activities
The assessment shall at least:

- Attempt control plane actions and resource access using an authenticated account that has not been explicitly granted authorization for those actions and verify that access is denied
- Confirm that access is denied by default unless explicitly granted
- Confirm that explicitly authorized accounts can perform permitted actions

5. Assessment verdict

- Pass: a default-deny policy is enforced for CO control plane actions and resource access; actions not explicitly authorized are denied

- Fail: a default-permit policy is applied; or control plane actions can be performed without explicit authorization

6. Assessment evidence

- [CONFIG] CO authorization policy configuration enforcing default-deny
- [DESIGN] Documentation of CO default-deny authorization model
- [LOG] / [OBSERVATION] Evidence that unauthorized access attempts are denied

6.4.3.4.2 Assessment Case AC-CO-AUTHZ-002

1. Assessment reference
Requirement: REQ-CO-AUTHZ-002 (Elevated)

2. Assessment objective
Verify that the CO enforces granular authorization for workload and resource control operations, binding permitted actions to authenticated users or service accounts and enforcing least-privilege permissions by default. The assessment shall not require a specific authorization model, provided that the implemented model supports granular permissions and least-privilege enforcement. The implemented authorization model may be role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the implemented CO authorization model, including whether it is role-based, attribute-based, policy-based, capability-based, or an equivalent mechanism.
- Documentation identifying workload and resource control operations protected by the authorization model.
- Documentation identifying authenticated users, service accounts, permissions, policy rules, or equivalent authorization bindings.
- A test configuration with at least two authenticated users or service accounts having different permissions for representative workload and resource control operations.
- Tools or interfaces to attempt permitted and non-permitted workload and resource control operations using those authenticated users or service accounts.

4. Assessment activities
The assessment shall at least:

- Identify the CO workload and resource control operations covered by REQ-CO-AUTHZ-002.
- Review the implemented authorization model and confirm that permitted actions are bound to authenticated users or service accounts.
- Verify that least-privilege permissions are enforced by default.
- Attempt a permitted workload or resource control operation using an authenticated user or service account authorized for that action and verify that the action succeeds.
- Attempt the same workload or resource control operation using an authenticated user or service account not authorized for that action and verify that the action is denied.
- Verify that the evidence demonstrates granular authorization without requiring a specific authorization model.

5. Assessment verdict

- Pass: the CO enforces granular authorization for workload and resource control operations, binds permitted actions to authenticated users or service accounts, and enforces least-privilege permissions by default, regardless of whether the implemented model is role-based, attribute-based, policy-based, capability-based, or equivalent.

- Fail: the CO does not enforce granular authorization for workload or resource control operations; or permitted actions are not bound to authenticated users or service accounts; or least-privilege permissions are not enforced by default; or an authenticated user or service account can perform an action for which it is not authorized.

6. Assessment evidence

- [DESIGN] Documentation of the implemented CO authorization model and its binding of permissions to authenticated users or service accounts.
- [CONFIG] Authorization policy, role, attribute, rule, capability, or equivalent configuration showing permitted and non-permitted workload and resource control operations.
- [LOG] / [OBSERVATION] Evidence of permitted workload or resource control operations succeeding for authorized users or service accounts and non-permitted operations being denied for unauthorized users or service accounts.

6.4.3.4.3 Assessment Case AC-CO-AUTHZ-003

1. Assessment reference
Requirement: REQ-CO-AUTHZ-003 (Advanced)

2. Assessment objective
Verify that the CO enforces fine-grained, policy-driven, or context-aware authorization for workload and resource control operations, including runtime context, contextual policy evaluation, or integration with external identity, access-management, or policy decision systems where applicable to the corresponding requirement and where such integration is part of the declared product configuration.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CO fine-grained, policy-driven, or context-aware authorization model.
- Documentation identifying workload and resource control operations protected by the authorization model.
- Documentation identifying contextual policy inputs, runtime context, policy rules, policy decision points, or equivalent mechanisms supported by the CO, where applicable.
- Documentation identifying whether integration with external identity, access-management, or policy decision systems is part of the declared product configuration.
- A test configuration with representative authenticated users, service accounts, permissions, contextual attributes, runtime context, policy inputs, or external authorization integration, as applicable to the declared product configuration.

4. Assessment activities
The assessment shall at least:

- Review the documented fine-grained, policy-driven, or context-aware CO authorization model.
- Verify that the CO enforces the applicable authorization policy for representative workload and resource control operations.
- Where runtime-context or context-aware enforcement is supported or applicable, verify that authorization decisions take the documented runtime or contextual inputs into account.
- Where integration with external identity, access-management, or policy decision systems is part of the declared product configuration, verify that authorization decisions use the integrated system as documented.
- Attempt a protected CO operation under an authorized policy condition and verify that the operation succeeds.
- Attempt the same protected CO operation under an unauthorized policy condition and verify that the operation is denied.

5. Assessment verdict

- Pass: the CO enforces fine-grained, policy-driven, or context-aware authorization for workload and resource control operations, including runtime context, contextual policy evaluation, or external authorization integration where applicable to the requirement and declared product configuration.
- Fail: the CO does not enforce fine-grained, policy-driven, or context-aware authorization where required; or runtime-context, contextual policy evaluation, or external authorization integration is claimed but not implemented or not effective in the declared product configuration; or an unauthorized user, service account, or unauthorized policy condition allows execution of a protected CO operation.

6. Assessment evidence

- [DESIGN] Documentation of the fine-grained, policy-driven, or context-aware CO authorization model.
- [CONFIG] Authorization policy, contextual policy, runtime-context configuration, or external authorization integration configuration, as applicable.
- [LOG] / [OBSERVATION] Evidence that CO authorization decisions are enforced according to the documented policy and declared product configuration.

6.4.3.5 Assessment for Confidentiality and Data Protection

6.4.3.5.1 Assessment Case AC-CO-CONF-001

1. Assessment reference
Requirement: REQ-CO-CONF-001 (Basic)

2. Assessment objective
Verify that the CO ensures that control plane secrets, including authentication tokens, private keys, and service credentials, are not stored in plaintext and are accessible only through a controlled interface. Verify also that the CO supports either internal secrets store or integration with a Key Management System (KMS).

3. Assessment preparation
The assessment shall have access to:

- Documentation describing how CO control plane secrets are stored and how access is controlled
- Documentation describing whether the CO uses an internal secrets store, integration with a Key Management System (KMS), or supports both
- A test environment with CO control plane secrets present in the managed environment
- Tools to inspect secret storage and to attempt direct access to secret values

4. Assessment activities
The assessment shall at least:

- Inspect the storage of CO control plane secrets and verify that they are not stored in plaintext
- Confirm that access to secrets is mediated through a controlled interface and that direct access to secret values without authorization is prevented
- Attempt to retrieve a secret value without the required authorization and verify that the attempt is denied
- Confirm that the CO supports either internal secrets store or integration with a KMS, as documented for the assessed product configuration

5. Assessment verdict

- Pass: CO control plane secrets are not stored in plaintext; access is mediated through a controlled interface; unauthorized retrieval attempts are denied; and the CO supports either internal secrets store or integration with a KMS

- Fail: CO control plane secrets are stored in plaintext; or secrets are directly accessible without authorization; or the CO does not support either internal secrets store or integration with a KMS

6. Assessment evidence

- [CONFIG] Secret storage configuration for the CO control plane
- [DESIGN] Documentation of secret storage, access-control architecture, and support for internal secrets store or KMS integration
- [OBSERVATION] Evidence that secrets are not in plaintext and unauthorized access is denied

6.4.3.5.2 Assessment Case AC-CO-CONF-002

1. Assessment reference
Requirement: REQ-CO-CONF-002 (Elevated)

2. Assessment objective
Verify that the CO ensures control-plane traffic and orchestrator-managed communication channels are encrypted, and enforces configuration policies that prevent unencrypted communication.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing CO enforcement of encryption for control-plane traffic and orchestrator-managed channels
- Documentation of configuration policies preventing unencrypted communication
- A test environment with encryption enforced for control-plane traffic
- Tools to capture CO control-plane traffic and to test encryption enforcement policies

4. Assessment activities
The assessment shall at least:

- Capture CO control-plane traffic and verify that it is encrypted
- Attempt to configure a workload or managed channel to use unencrypted communication and verify that the CO policy prevents it
- Confirm that orchestrator-managed channels enforce encryption

5. Assessment verdict

- Pass: CO control-plane traffic is encrypted; policies preventing unencrypted communication are enforced; orchestrator-managed channels use encryption
- Fail: control-plane traffic is not encrypted; or unencrypted communication can be configured without policy enforcement

6. Assessment evidence

- [CONFIG] Encryption enforcement configuration for CO control-plane and managed channels
- [DESIGN] Documentation of CO encryption enforcement model
- [OBSERVATION] Encrypted control-plane traffic capture and evidence of rejected unencrypted channel configurations

6.4.3.5.3 Assessment Case AC-CO-CONF-003

1. Assessment reference
Requirement: REQ-CO-CONF-003 (Advanced)

2. Assessment objective
Verify that the CO supports scheduling and managing container workloads in platform-supported confidential execution environments, enforcing workload placement, isolation constraints, and associated security policies.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing CO support for scheduling workloads in platform-supported confidential execution environments
- Documentation of CO-enforced workload placement and isolation constraints for confidential workloads
- A test environment with a platform supporting confidential execution and the CO configured for it
- Tools to schedule a workload in a confidential environment and to verify placement and isolation constraints

4. Assessment activities
The assessment shall at least:

- Schedule a test workload using the CO interface for confidential execution and verify that it is placed in the confidential execution environment as configured
- Confirm that CO-enforced placement and isolation constraints for confidential workloads are applied
- Attempt to schedule a workload in a confidential environment with a configuration violating placement or isolation constraints and verify that the CO rejects it

5. Assessment verdict

- Pass: the CO supports scheduling workloads in confidential execution environments; placement and isolation constraints are enforced; non-compliant placements are rejected
- Fail: the CO does not support scheduling for confidential environments; or placement/isolation constraints are not enforced

6. Assessment evidence

- [CONFIG] CO configuration for confidential execution scheduling and placement constraints
- [DESIGN] Documentation of CO support for confidential workload scheduling and enforcement
- [PLT-CAP] Platform capability evidence for confidential execution
- [OBSERVATION] Evidence of workload placed in confidential environment with constraints enforced

6.4.3.6 Assessment for Secure Update

6.4.3.6.1 Assessment Case AC-CO-UPD-001

1. Assessment reference
Requirement: REQ-CO-UPD-001 (Basic)

2. Assessment objective
Verify that:

- The CO is delivered in a form that supports authenticity and integrity verification of CO updates before installation or activation.
- The CO supports authenticity and integrity verification of CO updates using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Where the CO includes the mechanism that verifies, installs, or activates its updates, this mechanism prevents installation or activation of an update when authenticity or integrity verification fails and records a security-relevant event.

- Where update delivery, verification, installation, or activation is performed by a Host OS update service, package manager, platform update service, appliance update mechanism, orchestration-based update pipeline, or other Operational Environment mechanism outside the declared CO product or outside the responsibility of the manufacturer, the product security documentation identifies that mechanism, the expected security properties, and the assumptions required for secure update of the CO.

3. Assessment preparation

The assessment shall have access to:

- Documentation identifying the CO components delivered as part of the declared CO product.
- Documentation describing the update artefacts applicable to those CO components.
- Documentation describing how authenticity and integrity verification of CO updates is supported before installation or activation, including the verification metadata used, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Documentation identifying whether update delivery, verification, installation, or activation is performed by the CO, by another component under the responsibility of the manufacturer, or by an Operational Environment mechanism.
- Where the CO includes the mechanism that verifies, installs, or activates updates, a test configuration allowing use of a valid update artefact and an update artefact with invalid, missing, revoked, or intentionally modified authenticity or integrity protection, or an equivalent test condition.
- Where update delivery, verification, installation, or activation is performed by an Operational Environment mechanism, product security documentation identifying the expected update mechanism, expected security properties, and assumptions required for secure update of the CO.

4. Assessment activities

The assessment shall at least:

- Identify the CO components and update artefacts covered by REQ-CO-UPD-001.
- Check that the CO is delivered in a form that supports authenticity and integrity verification of CO updates before installation or activation.
- Check that the update artefacts support authenticity and integrity verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- Check whether update delivery, verification, installation, or activation is performed by the CO, by another component under the responsibility of the manufacturer, or by an Operational Environment mechanism.
- Where the CO includes the mechanism that verifies, installs, or activates updates, attempt to install or activate an update whose authenticity or integrity verification fails and verify that installation or activation is prevented and that a security-relevant event is recorded.
- Where update delivery, verification, installation, or activation is performed outside the declared CO product or outside the responsibility of the manufacturer, verify that the product security documentation identifies the Operational Environment dependency, the expected security properties, and the assumptions required for secure update of the CO.

5. Assessment verdict

- Pass: the CO is delivered in a form that supports authenticity and integrity verification of CO updates before installation or activation; update artefacts support verification using digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata; and, where the CO includes the mechanism that verifies, installs, or activates updates, failed authenticity or integrity verification prevents installation or activation and is recorded as a security-relevant event. Where update delivery, verification, installation, or activation is performed outside the declared CO product or outside the responsibility of the manufacturer, the Operational Environment dependency, expected security properties, and assumptions are documented.

- Fail: the CO is not delivered in a form that supports authenticity and integrity verification of CO updates before installation or activation; or the update artefacts do not support the required verification metadata; or, where the CO includes the mechanism that verifies, installs, or activates updates, an update whose authenticity or integrity verification fails can be installed or activated, or the failure is not recorded as a security-relevant event; or the relevant Operational Environment dependency is not documented where update delivery, verification, installation, or activation is performed outside the declared CO product or outside the responsibility of the manufacturer.

6. Assessment evidence

- [DESIGN] Documentation of the CO update model, including update artefacts and the component or mechanism responsible for update delivery, verification, installation, or activation.
- [DESIGN] Documentation of authenticity and integrity verification metadata used for CO updates, such as digital signatures, certificates, cryptographic digests, trusted reference values, or equivalent metadata.
- [CONFIG] Configuration of the CO update mechanism, where the CO includes the mechanism that verifies, installs, or activates updates.
- [LOG] / [OBSERVATION] Where the CO includes the mechanism that verifies, installs, or activates updates, evidence that a valid update is accepted and that an update with failed authenticity or integrity verification is prevented from installation or activation and recorded as a security-relevant event.
- [GUIDANCE] / [OE] Where update delivery, verification, installation, or activation is performed outside the declared CO product or outside the responsibility of the manufacturer, product security documentation identifying the Operational Environment dependency, expected security properties, and assumptions required for secure update of the CO.

6.4.3.6.2 Assessment Case AC-CO-UPD-002

1. Assessment reference

Requirement: REQ-CO-UPD-002 (Elevated)

2. Assessment objective

Verify that:

- The CO supports rollback protection for CO updates where rollback protection is provided by the CO or by an update mechanism under the responsibility of the manufacturer.
- Where rollback protection is supported, unauthorized installation of outdated, revoked, or replayed CO updates is prevented.
- Authorized rollback to a previously trusted CO version is permitted only when:
  a) the rollback image or package is verified for integrity and authenticity;
  b) the rollback action is explicitly authorized by an administrator with appropriate privileges or an equivalent authorized control function; and
  c) the rollback operation is recorded as a security-relevant event.

3. Assessment preparation
The assessment shall have access to:

- Documentation identifying whether rollback protection for CO updates is provided by the CO or by an update mechanism under the responsibility of the manufacturer.
- Documentation describing the rollback protection mechanism, where rollback protection is provided.
- Documentation describing procedures for authorized rollback to a previously trusted CO version, where authorized rollback is supported.

- Documentation identifying the administrator privileges or equivalent authorized control function required to authorize rollback, where authorized rollback is supported.
- Documentation describing how rollback operations are recorded as security-relevant events, where rollback is supported.
- A test or demonstration configuration allowing an outdated, revoked, or replayed CO update to be attempted, where rollback protection is provided by the CO or by an update mechanism under the responsibility of the manufacturer.
- A test or demonstration configuration allowing authorized rollback to a previously trusted CO version, where authorized rollback is supported.

4. Assessment activities

The assessment shall at least:

- Check whether rollback protection for CO updates is provided by the CO or by an update mechanism under the responsibility of the manufacturer.
- Where rollback protection is not provided by the CO or by an update mechanism under the responsibility of the manufacturer, verify that the product security documentation identifies the update and rollback assumptions applicable to the declared product configuration.
- Where rollback protection is provided, attempt to install an outdated, revoked, or replayed CO update outside the documented authorized rollback procedure and verify that installation is prevented.
- Where authorized rollback is supported, perform an authorized rollback using the documented procedure and verify that:
  a) the rollback image or package is verified for integrity and authenticity;
  b) the rollback action requires explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; and
  c) the rollback operation is recorded as a security-relevant event.

5. Assessment verdict

- Pass: where rollback protection is provided by the CO or by an update mechanism under the responsibility of the manufacturer, unauthorized installation of outdated, revoked, or replayed CO updates is prevented; and authorized rollback to a previously trusted CO version is permitted only following integrity and authenticity verification of the rollback image or package, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function, and recording of the rollback operation as a security-relevant event. Where rollback protection is not provided by the CO or by an update mechanism under the responsibility of the manufacturer, the update and rollback assumptions applicable to the declared product configuration are documented.
- Fail: where rollback protection is provided by the CO or by an update mechanism under the responsibility of the manufacturer, outdated, revoked, or replayed CO updates can be installed without authorization; or authorized rollback can be performed without integrity and authenticity verification; or rollback can be performed without explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function; or the rollback operation is not recorded as a security-relevant event.

6. Assessment evidence

- [DESIGN] Documentation identifying whether rollback protection is provided by the CO or by an update mechanism under the responsibility of the manufacturer.
- [DESIGN] Documentation of rollback protection mechanisms and authorized rollback procedures, where rollback protection or authorized rollback is supported.
- [CONFIG] Configuration or policy information showing rollback authorization controls, where authorized rollback is supported.

- [LOG] / [OBSERVATION] Where rollback protection is provided, evidence that outdated, revoked, or replayed updates are prevented outside the authorized rollback procedure.
- [LOG] / [OBSERVATION] Where authorized rollback is supported, evidence that rollback is performed only after integrity and authenticity verification, explicit authorization by an administrator with appropriate privileges or an equivalent authorized control function and recording as a security-relevant event.
- [GUIDANCE] / [OE] Where rollback protection is not provided by the CO or by an update mechanism under the responsibility of the manufacturer, documentation identifying update and rollback assumptions applicable to the declared product configuration.

6.4.3.6.3 Assessment Case AC-CO-UPD-003

1. Assessment reference
Requirement: REQ-CO-UPD-003 (Advanced)

2. Assessment objective
Verify that where the CO supports execution of update actions across managed cluster components as part of the declared product, it provides, for each managed component, identification of the update action, verification of update outcome, and detection of failed or partial rollout states.

3. Assessment preparation
The assessment shall have access to:

- Product security documentation identifying whether execution of update actions across managed cluster components is supported as part of the declared product.
- Documentation describing how the CO executes and tracks update actions across managed cluster components, where such update execution is supported.
- A test environment with the CO managing at least two cluster components, only where execution of update actions across managed cluster components is supported as part of the declared product.
- Tools to monitor per-component update execution and outcome records, where such update execution is supported.

4. Assessment activities
The assessment shall at least:

- Check whether execution of update actions across managed cluster components is supported as part of the declared product.
- If execution of update actions across managed cluster components is not supported as part of the declared product, check that this is identified in the product security documentation and proceed to the verdict based on the condition in REQ-CO-UPD-003 not being triggered.
- Where execution of update actions across managed cluster components is supported, initiate an update action across at least two managed cluster components and observe execution.
- Confirm that each managed component's update action is individually identified.
- Confirm that the update outcome for each managed component is verified and recorded.
- Simulate a failed or partial update on at least one managed component and verify that the CO detects and records the failed or partial rollout state.

5. Assessment verdict

- Pass: execution of update actions across managed cluster components is not supported as part of the declared product and this is documented; or, where such execution is supported, the CO identifies update actions per managed component, verifies update outcomes per managed component, and detects failed or partial rollout states.

- Fail: the product security documentation does not identify whether execution of update actions across managed cluster components is supported as part of the declared product; or execution of update actions across managed cluster components is supported as part of the declared product, but update actions are not identified per managed component; or update outcomes are not verified per managed component; or failed or partial rollout states are not detected.

6. Assessment evidence

- [DESIGN] Product security documentation identifying whether execution of update actions across managed cluster components is supported as part of the declared product.
- [DESIGN] Documentation of per-component update tracking, outcome verification, and failed or partial rollout detection, where such update execution is supported.
- [CONFIG] Multi-component update execution and outcome tracking configuration, where such update execution is supported.
- [LOG] / [OBSERVATION] Per-component update records showing update-action identification, outcome verification, and detection of failed or partial rollout states, where such update execution is supported.

6.4.3.7 Assessment for Decommissioning and Secure Transfer

6.4.3.7.1 Assessment Case AC-CO-DECOM-001

1. Assessment reference
Requirement: REQ-CO-DECOM-001 (Basic - Single class)

2. Assessment objective
Verify that the CO provides a mechanism to remove workload specifications, orchestration configuration, policy objects, and associated metadata managed by the CO, such that this data is no longer retrievable or usable through the interfaces and functions provided by the CO.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the CO removal mechanism and the workload specifications, orchestration configuration, policy objects, and associated metadata within its scope.
- A test configuration with assessed CO-managed data accessible through CO interfaces and functions before removal.
- Access to the CO interfaces and functions normally used to manage or retrieve that data.

4. Assessment activities
The assessment shall at least:

1) Review documentation to identify the removal mechanism and the workload specifications, orchestration configuration, policy objects, and associated metadata managed by the CO for the assessment.
2) Confirm, before removal, that the assessed data is retrievable or usable through the interfaces and functions provided by the CO.
3) Invoke the documented removal mechanism.
4) Verify, after removal, that the assessed data is no longer retrievable or usable through the interfaces and functions provided by the CO.

5. Assessment verdict

- Pass: The CO provides a removal mechanism and, after its invocation, the assessed workload specifications, orchestration configuration, policy objects, and associated metadata are no longer retrievable or usable through the interfaces and functions provided by the CO.

- Fail: No removal mechanism is provided; or, after removal, the assessed data remains retrievable or usable through the interfaces and functions provided by the CO.

6. Assessment evidence

- [DESIGN] Documentation of the CO removal mechanism and associated data within scope.
- [SCOPE] Identification of the assessed workload specifications, orchestration configuration, policy objects, and associated metadata.
- [CONFIG] Relevant CO configuration for the assessed removal procedure.
- [LOG] / [OBSERVATION] Evidence showing the accessible state before removal and the non-retrievable / non-usable state after removal through the interfaces and functions provided by the CO.

6.4.3.7.2 Assessment Case AC-CO-DECOM-002

1. Assessment reference
Requirement: REQ-CO-DECOM-002 (Basic - Single class)

2. Assessment objective
Verify that, where the CO supports transfer of workload specifications, orchestration configuration, policy objects, or associated metadata to another product or system, the CO protects the confidentiality and integrity of the transferred data through the interfaces and functions provided by the CO.

3. Assessment preparation
The assessment shall have access to:

- Documentation describing the supported transfer mechanism and the types of workload specifications, orchestration configuration, policy objects, or associated metadata that can be transferred.
- Documentation describing the security protections applied to the transfer.
- A test configuration in which supported data can be transferred from the CO to another product or system.
- Access to the interfaces and functions provided by the CO for the transfer operation.

4. Assessment activities
The assessment shall at least:

1) Review documentation to identify the supported transfer mechanism and the security protections applied to the transferred data.
2) Invoke the documented transfer mechanism for supported workload specifications, orchestration configuration, policy objects, or associated metadata.
3) Verify that the transfer is performed through the interfaces and functions provided by the CO with protection of the confidentiality and integrity of the transferred data.
4) Where feasible, verify that transfer without the documented protection is not permitted through the interfaces and functions provided by the CO.

5. Assessment verdict

- Pass: Where the CO supports transfer, the documented transfer mechanism protects the confidentiality and integrity of the transferred data through the interfaces and functions provided by the CO.
- Fail: A supported transfer mechanism does not protect the confidentiality or integrity of the transferred data; or transfer can be performed through the interfaces and functions provided by the CO without the documented protection.

6. Assessment evidence

- [DESIGN] Documentation of the transfer mechanism and associated security protections.

- [SCOPE] Identification of the transferred data types assessed.
- [CONFIG] Relevant CO configuration for the assessed transfer procedure.
- [LOG] / [OBSERVATION] Evidence showing the protected transfer of the assessed data through the interfaces and functions provided by the CO.

6.5 Assessment for Assurance Security Requirements

6.5.1 Assessment for Vulnerability Management

6.5.1.1 Assessment Case AC-ASSUR-VULM-001

1. Assessment Reference
Requirement: REQ-ASSUR-VULM-001

2 Assessment Objective
Verify that:

1) Known exploitable vulnerabilities affecting the product are identified.
2) Each identified known exploitable vulnerability is mitigated by appropriate technical or operational measures that sufficiently prevent or reduce the feasibility of exploitation in the product context, taking into account the intended purpose and reasonably foreseeable use of the product.
3) No identified known exploitable vulnerability affecting the product remains unmitigated.

3. Assessment Preparation
The assessment shall have access to:

- Product documentation identifying elements contained in the product.
- Component inventory, bill of materials, or equivalent identification information, including relevant version and patch-level information where available.
- Access to the product, or to relevant components, sufficient to perform vulnerability scanning where technically feasible.
- Vulnerability scanning tools and associated vulnerability databases suitable for identifying candidate vulnerabilities in the product.
- Access to recognized public vulnerability sources:
  - Public databases, e.g. EUVD, GCVE.EU, CISA and GCVE.EU Known Exploited Vulnerabilities (KEV) Catalog, the CVE List, the NIST National Vulnerability Database (NVD), and relevant vendor advisories.
  - Complementary sources that may support identification of relevant vulnerabilities, where relevant, for example technical research papers, conference papers, and other publicly available security research.
- Vulnerability assessment records and conclusions produced in accordance with prEN 40000-1-3 [i.13].
- Product guidance, where mitigation relies on measures described in such guidance.

4. Assessment Activities

1) Review the product documentation, inventory, bill of materials, or equivalent information to identify the elements contained in the product and their relevant versions or patch levels, where available.
2) Perform vulnerability scanning, where technically feasible, on the product or relevant components to identify candidate vulnerabilities affecting product elements.

3) Assess, in accordance with prEN 40000-1-3 [i.13], the vulnerabilities identified through correlation of scanning results, product identification information, vendor advisories, and recognized public vulnerability sources, in order to determine which of them are known exploitable vulnerabilities affecting the product.

4) For each identified known exploitable vulnerability, review the vulnerability assessment and the associated mitigation measures applied to the product.

5) Verify whether the applied mitigation sufficiently prevents or reduces the feasibility of exploitation in the product context, taking into account the intended purpose and reasonably foreseeable use of the product.

6) Where mitigation relies on measures described in the product guidance, review the guidance and verify that it clearly describes the relevant conditions, configurations, or operational measures.

7) Where the sufficiency of mitigation is not evident from documentation and analysis alone, perform additional inspection, configuration review, or targeted validation testing.

8) Verify that no identified known exploitable vulnerability affecting the product remains unmitigated.

5. Assessment Verdict

- Pass: Known exploitable vulnerabilities affecting the product are identified, and each such vulnerability is mitigated by appropriate technical or operational measures that sufficiently prevent or reduce the feasibility of exploitation in the product context.
- Fail: Vulnerability assessment is missing or insufficient, mitigation is missing or insufficient, product guidance is inadequate where relied upon, or one or more known exploitable vulnerabilities remain unmitigated.

6. Assessment Evidence

- [ANALYSIS] Vulnerability scanning results.
- [ANALYSIS] Vulnerability assessment records and conclusions produced in accordance with prEN 40000-1-3 [i.13].
- [GUIDANCE] Product guidance describing operational mitigations where relevant.
- [OBSERVATION] Assessment record showing the disposition of each identified known exploitable vulnerability.

6.5.2 Assessment for Software Bill of Materials (SBOM)

6.5.2.1 Assessment Case AC-ASSUR-SBOM-001

1. Assessment reference
Requirement: REQ-ASSUR-SBOM-001

2. Assessment objective

- Verify that the SBOM provided for the VES or CES product:
- covers the in-scope components of the VES or CES as described in clauses 4.2.2.2 and 4.2.3.1;
- identifies the software components and dependencies included in the product;
- identifies the names and versions of those software components and dependencies;
- includes supplier or origin information where available;
- includes Software Heritage persistent IDentifiers (SWHIDs) for open-source components where available;
- is provided in a machine-readable format.

3. Assessment preparation
The assessment shall have access to:

- the SBOM provided for the VES or CES product;
- product architecture documentation identifying the in-scope VES or CES components;
- documentation or metadata identifying software components and dependencies included in the product;
- supplier or origin information where available;
- SWHIDs for open-source components where available;
- information identifying the machine-readable SBOM format.

4. Assessment activities
The assessment shall at least:

- Check that the SBOM covers the in-scope components of the VES or CES as described in clauses 4.2.2.2 and 4.2.3.1.
- Check that the SBOM identifies software components included in the product.
- Check that the SBOM identifies dependencies included in the product.
- Check that the SBOM identifies names and versions of the software components and dependencies.
- Check that supplier or origin information is included where available.
- For open-source components, check that SWHIDs are included where available.
- Check that the SBOM is provided in a machine-readable format.

5. Assessment verdict

- Pass: the SBOM covers the in-scope VES or CES components and identifies software components, dependencies, names, versions, supplier or origin information where available, SWHIDs for open-source components where available, and is provided in a machine-readable format.
- Fail: the SBOM is missing or does not cover the in-scope VES or CES components, or does not identify the required software component, dependency, version, supplier or origin information, or SWHID information required by REQ-ASSUR-SBOM-001; or the SBOM is not provided in a machine-readable format.

6. Assessment evidence

- [DOCUMENT] SBOM for the VES or CES product.
- Evidence of software components and dependencies, names and versions.
- Supplier or origin information where available.
- SWHIDs for open-source components where available.
- Machine-readable SBOM format evidence.
- [DESIGN] Product architecture documentation identifying in-scope components.

6.5.2.2 Assessment Case AC-ASSUR-SBOM-002

1. Assessment reference
Requirement: REQ-ASSUR-SBOM-002

2. Assessment objective
Verify that, for VES products, the SBOM includes, at minimum:

- hypervisor executable artefacts, such as the Hypervisor image, kernel, microkernel, virtual machine monitor, or equivalent executable components;
- hypervisor management interfaces, such as CLI or API components;
- Virtual Machine Monitor (VMM) and lifecycle controllers, where applicable;
- resource scheduling components for CPU, memory, or I/O, where included in the product;
- device virtualisation modules;
- software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the VES product.

3. Assessment preparation
The assessment shall have access to:

- the SBOM for the VES product;
- product architecture documentation identifying in-scope VES components;
- documentation identifying Hypervisor executable artefacts and other VES components included in the product, including software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the VES product, where applicable.

4. Assessment activities
The assessment shall at least:

- Check that the SBOM includes Hypervisor executable artefacts, such as the Hypervisor image, kernel, microkernel, virtual machine monitor, or equivalent executable components.
- Check that the SBOM includes Hypervisor management interfaces, such as CLI or API components.
- Where applicable, check that the SBOM includes VMM and lifecycle controllers.
- Where included in the product, check that the SBOM includes resource scheduling components for CPU, memory, or I/O.
- Check that the SBOM includes device virtualisation modules.
- Check that the SBOM includes software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the VES product, where applicable.

5. Assessment verdict

- Pass: the SBOM includes the VES component categories required by REQ-ASSUR-SBOM-002, including software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the VES product, where applicable.
- Fail: the SBOM does not include one or more VES component categories required by REQ-ASSUR-SBOM-002 and the omission is not justified by the declared product architecture.

6. Assessment evidence

- [DOCUMENT] SBOM for the VES product.
- Evidence that the SBOM includes the VES component categories required by REQ-ASSUR-SBOM-002, including software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the VES product, where applicable.
- [DESIGN] Product architecture documentation identifying in-scope VES components.

6.5.2.3 Assessment Case AC-ASSUR-SBOM-003

1. Assessment reference
Requirement: REQ-ASSUR-SBOM-003

2. Assessment objective
Verify that, for CES products, the SBOM includes, at minimum:

- CRS executable artefacts, such as runtime binaries, runtime daemons, runtime shims, enforcement modules, APIs, or services;
- Container Engine components, such as engine binaries, image management tools, container networking modules, storage or volume plugins, and supported extensions;
- Container Orchestrator components, such as orchestrator binaries, admission controllers, scheduling modules, secrets management, network policy enforcement, and cluster lifecycle management components;
- Software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the CES product.

3. Assessment preparation
The assessment shall have access to:

- the SBOM for the CES product;
- product architecture documentation identifying in-scope CES components;
- documentation identifying CRS, CE, CO, and software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the CES product, where applicable.

4. Assessment activities
The assessment shall at least:

- Check that the SBOM includes CRS executable artefacts, such as runtime binaries, runtime daemons, runtime shims, enforcement modules, APIs, or services.
- Check that the SBOM includes Container Engine components, such as engine binaries, image management tools, container networking modules, storage or volume plugins, and supported extensions.
- Check that the SBOM includes Container Orchestrator components, such as orchestrator binaries, admission controllers, scheduling modules, secrets management, network policy enforcement, and cluster lifecycle management components.
- Check that the SBOM includes software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the CES product, where applicable.

5. Assessment verdict

- Pass: the SBOM includes the CES component categories required by REQ-ASSUR-SBOM-003, including software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the CES product, where applicable.
- Fail: the SBOM does not include one or more CES component categories required by REQ-ASSUR-SBOM-003 and the omission is not justified by the declared product architecture.

6. Assessment evidence

- [DOCUMENT] SBOM for the CES product.
- Evidence that the SBOM includes the CES component categories required by REQ-ASSUR-SBOM-003, including software, firmware, drivers, libraries, packages, or other digital artefacts associated with hardware-related dependencies and bundled with the CES product, where applicable.
- [DESIGN] Product architecture documentation identifying in-scope CES components.

Annex A (informative): Relationship between the present document and the requirements of EU Regulation (EU) 2024/2847 - the Cyber Resilience Act

The present document has been prepared under the Commission's standardisation request C(2025) 618 [i.3] final to provide one voluntary means of conforming to the requirements of Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) No 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) (CRA) [i.1].

Once the present document is cited in the Official Journal of the European Union under that Regulation, compliance with the normative clauses of the present document given in table A.1 confers, within the limits of the scope of the present document, a presumption of conformity with the corresponding requirements of that Regulation and associated EFTA regulations.

Table A.1: Relationship between the present document and the requirements of Regulation (EU) 2024/2847 - the Cyber Resilience Act

Description	Essential Requirements of Regulation (EU) 2024/2847	Clause(s) of the present document	Remarks / Notes
Annex I, Part I, (1)	*Products with digital elements shall be designed, developed and produced in such a way that they ensure an appropriate level of cybersecurity based on the risks.*	Clause 5.1; clauses 5.2.1, 5.2.2, 5.3.1, 5.3.2, 5.3.3, 5.4.1 and 5.4.2; corresponding assessment provisions in clause 6; Annex K where cryptographic mechanisms are used; Annex R where applicable.	Clause 5.1 defines the applicability of requirements according to the declared product boundary, use case and Security Profile. Clauses 5.2 and 5.3 define product-specific requirements for VES and CES components. Clause 5.4 defines the assurance security requirements retained in the present document. Clause 6 defines the corresponding assessment provisions. Annex K applies to cryptographic mechanisms used by product security functions. Annex R applies only where the product relies on an RDPS in accordance with clause 5.1. Annex B provides the informative security analysis supporting the use cases, risk levels and Security Profiles.
Annex I, Part I, (2)(a)	*Products with digital elements shall be made available on the market without known exploitable vulnerabilities.*	Clause 5.4.1; corresponding assessment provisions in clause 6.5.1.	Covered by REQ-ASSUR-VULM-001. The requirement is limited to the product-specific outcome that the product does not contain unmitigated known exploitable vulnerabilities.
Annex I, Part I, (2)(b)	*Products with digital elements shall be made available on the market with a secure by default configuration, unless otherwise agreed between manufacturer and business user in relation to a tailor-made product with digital elements, including the possibility to reset the product to its original state.*	Clauses 5.2.1.9, 5.2.2.4, 5.3.1.9, 5.3.2.1 and 5.3.3.2; corresponding assessment provisions in clause 6.	Covered by the secure configuration and default requirements for the Hypervisor, M&O System and CRS, and by relevant CE and CO requirements addressing default isolation and integrity of configuration-relevant mechanisms. Applies according to the declared product boundary, relevant component and applicable Security Profile.

Description	Essential Requirements of Regulation (EU) 2024/2847	Clause(s) of the present document	Remarks / Notes
Annex I, Part I, (2)(c)	*Products with digital elements shall ensure that vulnerabilities can be addressed through security updates, including, where applicable, through automatic security updates that are installed within an appropriate timeframe enabled as a default setting, with a clear and easy-to-use opt-out mechanism, through the notification of available updates to users, and the option to temporarily postpone them.*	Clauses 5.2.1.8, 5.2.2.8, 5.3.1.8, 5.3.2.6, 5.3.3.6 and 5.4.1; corresponding assessment provisions in clause 6.	Covered by the secure update requirements for the Hypervisor, M&O System, CRS, CE and CO, and by REQ-ASSUR-VULM-001 for the product-specific outcome related to known exploitable vulnerabilities. Process-oriented vulnerability-handling requirements are not duplicated in the present vertical standard.
Annex I, Part I, (2)(d)	*Products with digital elements shall ensure protection from unauthorised access by appropriate control mechanisms, including but not limited to authentication, identity or access management systems, and report on possible unauthorised access.*	Clauses 5.2.1.3, 5.2.1.4, 5.2.1.7, 5.2.2.2, 5.2.2.3, 5.2.2.7, 5.3.1.3, 5.3.1.4, 5.3.1.7, 5.3.2.3, 5.3.2.4, 5.3.3.3 and 5.3.3.4; clauses R.4 and R.5 where Annex R applies; corresponding assessment provisions in clause 6.	Covered by authentication, authorization and logging requirements for the Hypervisor, M&O System, CRS, CE and CO. Where Annex R applies, supplementary RDPS-boundary authentication requirements apply to the local product side and, where applicable, the RDPS side.
Annex I, Part I, (2)(e)	*Products with digital elements shall protect the confidentiality of stored, transmitted or otherwise processed data, personal or other, such as by encrypting relevant data at rest or in transit by state-of-the-art mechanisms, and by using other technical means.*	Clauses 5.2.1.5, 5.2.2.5, 5.3.1.5, 5.3.2.5 and 5.3.3.5; Annex K; clauses R.4 and R.5 where Annex R applies; corresponding assessment provisions in clause 6.	Covered by confidentiality protection requirements for the Hypervisor, M&O System, CRS, CE and CO, including protection of data at rest, data in transit and, where applicable, data in use or confidential execution mechanisms. Annex K applies to cryptographic mechanisms used for confidentiality protection. Where Annex R applies, supplementary RDPS-boundary confidentiality requirements apply.
Annex I, Part I, (2)(f)	*Products with digital elements shall protect the integrity of stored, transmitted or otherwise processed data, personal or other, commands, programs and configuration against any manipulation or modification not authorised by the user, and report on corruptions.*	Clauses 5.2.1.2, 5.2.1.7, 5.2.2.5, 5.2.2.6, 5.3.1.2, 5.3.1.7, 5.3.2.2 and 5.3.3.2; Annex K; clauses R.4 and R.5 where Annex R applies; corresponding assessment provisions in clause 6.	Covered by boot integrity verification, image integrity verification, runtime integrity protection, remote attestation, communication integrity, M&O integrity protection, CE integrity protection, CO integrity protection and logging requirements. Annex K applies to cryptographic mechanisms used for integrity protection. Where Annex R applies, supplementary RDPS-boundary integrity requirements apply.
Annex I, Part I, (2)(g)	*Products with digital elements shall process only data, personal or other, that are adequate, relevant and limited to what is necessary in relation to the intended purpose of the product with digital elements (data minimisation).*	Clauses 5.2.1.10 and 5.3.1.10; corresponding assessment provisions in clause 6.	Covered by the data minimisation requirements for the Hypervisor and CRS. Applies where telemetry, attestation evidence, runtime data, logging data or other product data are processed by the in-scope product components.
Annex I, Part I, (2)(h)	*Products with digital elements shall protect the availability of essential and basic functions, also after an incident, including through resilience and mitigation measures against denial-of-service attacks.*	Clauses 5.2.1.1, 5.2.1.6, 5.3.1.1 and 5.3.1.6; clauses R.4 and R.5 where Annex R applies; corresponding assessment provisions in clause 6.	Covered by isolation, resource control, availability and resilience requirements for the Hypervisor and CRS. Where Annex R applies, supplementary RDPS-boundary availability requirements apply to the RDPS-dependent product function.

Description	Essential Requirements of Regulation (EU) 2024/2847	Clause(s) of the present document	Remarks / Notes
Annex I, Part I, (2)(i)	*Products with digital elements shall minimise the negative impact by the products themselves or connected products on the availability of services provided by other devices or networks.*	Clauses 5.2.1.1, 5.2.1.6, 5.3.1.1, 5.3.1.6 and 5.3.3.1; corresponding assessment provisions in clause 6.	Covered by workload isolation, network plane separation, resource governance, availability, resilience and orchestrator isolation requirements. Applies according to the relevant VES or CES use case, declared product boundary and Security Profile.
Annex I, Part I, (2)(j)	*Products with digital elements shall be designed, developed and produced to limit attack surfaces, including external interfaces.*	Clauses 5.2.1.1, 5.2.1.9, 5.2.2.4, 5.2.2.5, 5.3.1.1, 5.3.1.9, 5.3.2.1 and 5.3.3.1; clauses R.4 and R.5 where Annex R applies; corresponding assessment provisions in clause 6.	Covered by control plane isolation, network plane separation, secure configuration, communication security, CE isolation and CO isolation requirements. Where Annex R applies, supplementary RDPS-boundary requirements address the product-facing RDPS boundary as an additional attack surface.
Exploitation mitigation and incident impact reduction	*Annex I, Part I, (2)(k): Products with digital elements shall be designed, developed and produced to reduce the impact of an incident using appropriate exploitation mitigation mechanisms and techniques.*	Clauses 5.2.1.1, 5.2.1.5, 5.2.1.6, 5.2.1.8, 5.3.1.1, 5.3.1.5, 5.3.1.6, 5.3.2.5 and 5.3.3.5; clauses R.4 and R.5 where Annex R applies; corresponding assessment provisions in clause 6.	Covered by advanced isolation, confidentiality, availability, resilience and secure update requirements for VES and CES components. Where Annex R applies, RDPS availability requirements contribute to incident-impact reduction through degraded behaviour, secure state, recovery support and continuity mechanisms.
Annex I, Part I, (2)(l)	*Products with digital elements shall provide security-related information by recording and monitoring relevant internal activity, including the access to or modification of data, services or functions, with an opt-out mechanism for the user.*	Clauses 5.2.1.7, 5.2.2.7, 5.3.1.7, 5.2.1.10 and 5.3.1.10; corresponding assessment provisions in clause 6.	Covered by logging requirements for the Hypervisor, M&O System and CRS, and by data minimisation requirements related to security-related information. Applies according to the declared product boundary, applicable component and Security Profile.
Annex I, Part I, (2)(m)	*Products with digital elements shall provide the possibility for users to securely and easily remove on a permanent basis all data and settings and, where such data can be transferred to other products or systems, ensure that this is done in a secure manner.*	Clauses 5.2.1.11, 5.2.2.9, 5.3.1.11, 5.3.2.7 and 5.3.3.7; corresponding assessment provisions in clause 6.	Covered by decommissioning and secure transfer requirements for the Hypervisor, M&O System, CRS, CE and CO. Applies to removal or secure transfer of data, configuration, metadata and settings managed by the in-scope product components.
Annex I, Part II	*Vulnerability handling requirements*	Clauses 5.4.1 and 5.4.2; corresponding assessment provisions in clause 6.5.	Covered for the product-specific outcome of no unmitigated known exploitable vulnerabilities by REQ-ASSUR-VULM-001. Covered for SBOM by REQ-ASSUR-SBOM-001 to REQ-ASSUR-SBOM-003. Broader vulnerability-handling process requirements are not duplicated in the present vertical standard.

Key to columns:

Description A textual reference to the requirement.

Requirements of Regulation

Identification of point(s) defining the requirement in Regulation (EU) 2024/2847 - the Cyber Resilience Act.

Clause(s) of the present document

Identification of clause(s) addressing the requirement in the present document

Presumption of conformity stays valid only as long as a reference to the present document is maintained in the list published in the Official Journal of the European Union. Users of the present document should consult frequently the latest list published in the Official Journal of the European Union.

Other Union legislation may be applicable to the product(s) falling within the scope of the present document.

Annex B (informative): Security Analysis

B.1 Overview

This annex applies a state-of-the-art methodology to identify assets and threats, identify and evaluate risk factors, and define Security Profiles applicable to the different use cases identified in the product context.

This annex is informative and non-binding. It provides a risk assessment methodology for assessing the cybersecurity risk of a product based on its intended use case and deployment environment. This approach helps determine the risk level for various use cases, guiding the application of appropriate security requirements.

Manufacturers can use this methodology as guidance or apply their own established risk assessment methodologies.

The risk assessment methodology used herein is a semi-quantitative approach, combining qualitative analysis with numerical scoring. It provides a framework for:

- Identifying product cybersecurity threats.
- Determining the likelihood and magnitude of impact if those threats materialize.
- Evaluating the overall cybersecurity risk for the product within its specific use case and deployment environment.

This methodology aligns with the general principles for assessing product cybersecurity risk, as defined in prEN 40000-1-2 [i.11].

B.2 Threats

B.2.1 Overview

Table B.2.1-1 maps identified cybersecurity threats to the specific components they affect within the scope of the present document: Hypervisor Type I, Hypervisor Type II, Orchestration & Management System, Container Orchestrator, Container Engine, and Container Runtime System.

The first column lists the threat categories considered in the present document, grouped by their primary security impact, such as interference or escape, integrity compromise, unauthorized access, data disclosure, denial of service, logging issues, malicious updates, insecure configurations, and supply chain compromise.

Each threat is referenced by its unique Threat ID, ensuring traceability to its detailed description in the threat catalogue in the following clauses. This mapping provides a clear overview of which threats are applicable to each component type, serving as the foundation for the risk assessment and for defining the corresponding security requirements and controls.

Table B.2.1-1: Threats to components mapping

Threat	Hypervisor Type I	Hypervisor Type II	Orchestration & Management	Container Orchestrator	Container Engine	Container Runtime System
Interference / Escape	T-HYP1-INTF-ESC	T-HYP2-HOST-ABUSE	T-M&O-CTRL-PLANE	T-CO-INTF-ESC	T-CE-INTF-ESC	T-CRS-INTF-ESC
Integrity Compromise	T-HYP1-INTEG-COMP	T-HYP2-INTEG-COMP	T-M&O-INTEG-TRUST	T-CO-INTEG-TRUST	T-CE-INTEG-COMP	T-CRS-INTEG-COMP
Unauthorized Admin Access	T-HYP1-UNAUTH-ADMIN	T-HYP2-UNAUTH-ADMIN	T-M&O-UNAUTH-ADMIN	T-CO-UNAUTH-ADMIN	T-CE-UNAUTH-ADMIN	T-CRS-UNAUTH-ADMIN
Unauthorized Privileged Actions	T-HYP1-UNAUTH-FUNC	T-HYP2-UNAUTH-FUNC	T-M&O-UNAUTH-FUNC	T-CO-UNAUTH-FUNC	T-CE-UNAUTH-FUNC	T-CRS-UNAUTH-FUNC
Sensitive Data Disclosure	T-HYP1-DATA-DISC	T-HYP2-DATA-DISC	T-M&O-DATA-DISC	T-CO-DATA-DISC	T-CE-DATA-DISC	T-CRS-DATA-DISC
DoS via Resource Exhaustion	T-HYP1-AVAIL-DOS	T-HYP2-AVAIL-DOS	T-M&O-AVAIL-DOS	T-CO-AVAIL-DOS	T-CE-AVAIL-DOS	T-CRS-AVAIL-DOS
Insufficient / Tampered Logging	T-HYP1-LOG-INT	T-HYP2-LOG-INT	T-M&O-LOG-INT	T-CO-LOG-INT	T-CE-LOG-INT	T-CRS-LOG-INT
Malicious / Compromised Update	T-HYP1-UPD-COMP	T-HYP2-UPD-COMP	T-M&O-UPD-COMP	T-CO-UPD-COMP	T-CE-UPD-COMP	T-CRS-UPD-COMP
Insecure Default and Changed Configurations	T-HYP1-CFG-COMP	T-HYP2-CFG-COMP	T-M&O-CFG-COMP	T-CO-CFG-COMP	T-CE-CFG-COMP	T-CRS-CFG-COMP
Supply Chain Compromise	T-HYP1-SC-COMP	T-HYP2-SC-COMP	T-M&O-SC-COMP	T-CO-SC-COMP	T-CE-SC-COMP	T-CRS-SC-COMP
Compromise via hardware-level or platform trust components	T-ALL-HW-ACCESS					

B.2.2 VES

B.2.2.1 Hyper Type I

T-HYP1-INTF-ESC: Inter-VM interference and escape via hypervisor/host interfaces

A malicious VM exploits flaws in the hypervisor, device emulation, vSwitch, or shared hardware to read/alter another VM's memory/state or escape to the hypervisor/root.

T-HYP1-INTEG-COMP: Integrity compromise of workloads or platform components

A malicious actor tampers with hypervisor binaries, kernel modules, configuration files, or VM images on the host storage, injecting malicious code or disabling isolation/security controls. The compromised hypervisor continues to operate without detection by higher-level systems, allowing the attacker to manipulate or monitor workloads.

T-HYP1-UNAUTH-ADMIN: Unauthorized access to Type-1 hypervisor management interfaces

An attacker exploits missing or weak authentication on the hypervisor's direct management interfaces (e.g. vSphere Client, ESXi Shell, SSH, vendor-specific APIs). This may be due to default credentials, weak password policies, outdated authentication protocols, or disabled MFA.

T-HYP1-UNAUTH-FUNC: Unauthorized privileged actions or data access on a Type-1 hypervisor

An attacker with valid but limited credentials (e.g. VM operator role) or a compromised low-privileged API token exploits insufficient authorization enforcement on the hypervisor management plane to perform high-impact actions or access restricted data.

T-HYP1-DATA-DISC: Unauthorized disclosure of sensitive data in a Type-1 hypervisor environment

An attacker exploits vulnerabilities, weak encryption, or misconfigurations in the hypervisor or its management plane to gain unauthorized access to sensitive data from virtual machines or platform resources.

This may include data in use (e.g. reading contents directly from RAM or CPU registers of running workloads), data at rest (e.g. accessing virtual disk images or configuration files), and data in transit (e.g. intercepting network traffic between VMs).

T-HYP1-AVAIL-DOS: DoS via resource exhaustion in a Type-1 hypervisor environment

An attacker either from a malicious or compromised guest VM, or via external network/API access, consumes excessive CPU, memory, I/O bandwidth, or network capacity on the hypervisor host, leading to degraded performance or full unavailability of hosted VMs.

This also includes scenarios involving resource overcommitment, where CPU steal time, memory contention, or shared I/O bottlenecks impact the performance of co-located VMs, potentially leading to degraded availability even when individual VM resource limits are not exceeded.

T-HYP1-LOG-INT: Insufficient logging or log tampering in a Type-1 hypervisor environment

An attacker with access to a guest VM or hypervisor management interface exploits insufficient logging, disabled audit trails, or insecure log storage to hide malicious actions (e.g. VM configuration changes, hypervisor setting modifications). They may also tamper with existing logs to erase evidence of compromise.

T-HYP1-UPD-COMP: Malicious or compromised update in a Type-1 hypervisor environment

An attacker injects, replaces, or replays a hypervisor firmware or software update to gain control over the Virtualisation layer. Because Type-1 hypervisors run directly on hardware, compromise here provides full access to all guest VMs and host resources.

T-HYP1-CFG-COMP: Insecure or unauthorized configuration in a Type-1 Hypervisor

An attacker exploits insecure default settings or unauthorized/insecure configuration changes to the hypervisor (via CLI, web UI, API, or configuration files). This can weaken isolation, expose management services, or disable logging/security controls, leading to unauthorized access, privilege escalation, or compromise of workloads.

T-HYP1-SC-COMP: Supply chain compromise in a Type-1 Hypervisor

An attacker compromises a dependency used in the hypervisor's build process, such as a low-level driver, cryptographic library, or firmware, to inject a backdoor or exploitable vulnerability.

Because the malicious code is baked into the hypervisor's core, every VM running on the host inherits the exposure, giving the attacker a privileged foothold.

B.2.2.2 Hyper Type II

T-HYP2-HOST-ABUSE: Guest-to-host escalation via host OS dependency and shared services

A malicious VM abuses the host OS attack surface to access host resources or other VMs.

T-HYP2-INTEG-COMP: Integrity compromise in hosted Virtualisation

An attacker compromises the host OS or hypervisor application binaries/configuration, or tampers with VM images in the host filesystem. Because Type-2 relies on the host OS, compromise of the host environment compromises the Virtualisation layer and workloads.

T-HYP2-UNAUTH-ADMIN: Unauthorized access to Type-2 hypervisor or host OS admin functions

An attacker exploits weak authentication on the hypervisor application or on the host OS's administrative accounts. Since Type-2 hypervisors run on a general-purpose OS, compromise of the host admin account grants full control over the hypervisor and all guest VMs.

T-HYP2-UNAUTH-FUNC: Unauthorized privileged actions or data access on a Type-2 hypervisor

An attacker with authenticated but restricted access to the hypervisor application or underlying OS exploits inadequate authorization to perform actions reserved for administrators. Since Type-2 hypervisors rely on the host OS for resource access, weak separation can also allow escalation to host-level privileges.

T-HYP2-DATA-DISC: Unauthorized disclosure of sensitive data in a Type-2 hypervisor environment

An attacker leverages weaknesses in the hosted hypervisor or host OS to read sensitive data belonging to VMs or the platform.

This may include data in use (e.g. extracting secrets from process memory), data at rest (e.g. accessing VM image files stored on the host), and data in transit (e.g. capturing inter-VM or host-VM communications).

T-HYP2-AVAIL-DOS: DoS via resource exhaustion in a Type-2 hypervisor environment

An attacker running on a guest VM or the host OS consumes excessive system resources (CPU, memory, storage, network), causing other VMs or the hypervisor process to stall or crash.

T-HYP2-LOG-INT: Insufficient logging or log tampering in a Type-2 hypervisor environment

An attacker running on the host OS or inside a VM exploits weak logging practices to erase or manipulate event records (e.g. VM lifecycle events, admin actions) to avoid detection. In a Type-2 setup, logs are often stored in user-accessible locations on the host, making them easier to modify.

T-HYP2-UPD-COMP: Malicious or compromised update in a Type-2 hypervisor environment

An attacker injects or manipulates updates to the hypervisor application running on top of the host OS. Because these updates often come via user-initiated downloads, attackers may deliver installers that have been modified with malicious code or downgrade the software to a vulnerable version.

T-HYP2-CFG-COMP: Unauthorized or insecure configuration in a Type-2 hypervisor

A Type-2 hypervisor may be exposed to insecure configuration either through unauthorized or insecure modifications of its parameters via the host operating system environment, local malware, or its application interface, or through insecure defaults inherited from the host operating system or shipped by the hypervisor itself. Examples include permissive VM networking modes, host-guest shared folders enabled by default, or control channels without authentication. Such insecure or unauthorized configurations can weaken VM isolation, expose the host system, or allow privilege escalation between guest VMs and the host.

T-HYP2-SC-COMP: Supply chain compromise in a Type-2 Hypervisor

An attacker compromises a dependency used by the Type-2 hypervisor, such as a driver, shared library, system component, or packaged module, during development, upstream distribution, or packaging.

When the hypervisor is installed on a user's OS, the malicious dependency executes with host-level privileges, allowing the attacker to compromise both the hypervisor and the host OS.

B.2.2.3 Orchestration and Management system

T-M&O-CTRL-PLANE: Control-plane abuse leading to cross-workload interference and privilege escalation

An attacker with (or who obtains) control-plane/API access manipulates scheduling, networking, storage or policies to read/impact other tenants' workloads or to break isolation at scale.

T-M&O-INTEG-TRUST: Misplaced trust in compromised platform

An orchestrator and management system treats a hypervisor host as trustworthy without verifying its integrity (no attestation, or attestation ignored). A compromised host can feed falsified state/configuration data back to the orchestrator and run tampered workloads while appearing legitimate.

T-M&O-UNAUTH-ADMIN: Unauthorized access to orchestration/management plane

An attacker gains unauthorized access to the orchestration platform's administrative APIs, dashboards, or CLI tools through weak authentication, missing identity federation controls, or compromised admin accounts. This provides cluster-wide control of all connected hypervisors and workloads.

T-M&O-UNAUTH-FUNC: Unauthorized privileged actions or data access in orchestration/management systems

An attacker with authenticated but restricted access to the orchestration platform exploits insufficient authorization on APIs or dashboards to invoke privileged actions cluster-wide or access sensitive data.

T-M&O-DATA-DISC: Unauthorized disclosure of sensitive data via Virtualisation orchestration/management system

An attacker abuses vulnerabilities, excessive privileges, or misconfigured access controls in the orchestration platform to obtain sensitive data from managed VMs or platform resources.

This may include data in use (e.g. retrieving memory dumps from running workloads), data at rest (e.g. accessing stored VM snapshots or configuration files), and data in transit (e.g. intercepting orchestrator-mediated traffic).

T-M&O-AVAIL-DOS: DoS via resource exhaustion in Virtualisation orchestration/management systems

An attacker overwhelms the orchestration control plane, APIs, or schedulers with excessive requests, malformed workload definitions, or frequent provisioning/de-provisioning cycles. This can lead to unresponsiveness of management tools and inability to operate workloads.

T-M&O-LOG-INT: Insufficient logging or log tampering in Virtualisation orchestration/management systems

An attacker targets the orchestration system's audit and logging functions to hide or falsify records of administrative actions, API calls, or workload lifecycle events. These systems often serve as the central source of truth, so tampering can mislead both human operators and automated security processes.

T-M&O-UPD-COMP: Malicious or compromised update in Virtualisation orchestration/management systems

An attacker tampers with updates for the orchestration platform or its plugins/extensions. A compromised orchestration system can push malicious configurations or updates to all managed hypervisors and workloads, multiplying the impact.

T-M&O-CFG-COMP: Unauthorized or insecure configuration in Virtualisation orchestration/management systems

An attacker exploits insecure default settings (such as default administrator accounts, weak API tokens, unencrypted management channels, or overly permissive role assignments) or performs unauthorized modifications through the orchestration system's management interfaces or APIs. Because the orchestration system controls multiple hypervisors and cluster-wide policies, insecure defaults or misconfigurations can propagate across the entire environment, weakening isolation, exposing management services, or granting attackers elevated privileges.

T-M&O-SC-COMP: Supply chain compromise in Virtualisation orchestration/management systems

An attacker injects malicious code into a dependency (for example, a module, API library, or plugin) used in the orchestrator's control plane. Because orchestrators have administrative access to all managed hosts, the compromise cascades to the entire infrastructure once deployed.

B.2.3 CES

B.2.3.1 Container Orchestrator

T-CO-INTF-ESC: Orchestrator-level cross-workload interference

An attacker with access to orchestration APIs or permissions in one namespace/project abuses orchestration misconfigurations, policy gaps, or network segmentation flaws to interfere with workloads managed by the orchestrator.

T-CO-INTEG-TRUST: Orchestrator trusts compromised nodes or components

The orchestrator schedules workloads onto a container host (node) or engine whose binaries, configuration, or runtime state have been tampered with, without verifying its integrity. This may be due to the absence of attestation, or attestation being ignored or spoofed.

T-CO-UNAUTH-ADMIN: Unauthorized access to orchestrator management plane

An attacker gains access to the orchestrator's administrative interfaces (web dashboards, CLI tools, APIs) due to weak or missing authentication controls, default credentials, or compromised admin accounts. Since the orchestrator manages scheduling, scaling, and policy enforcement for all workloads, compromise can result in complete cluster control.

T-CO-UNAUTH-FUNC: Unauthorized privileged operations or data access in the orchestrator

An attacker with valid but low-privileged credentials or a compromised service account exploits insufficient authorization enforcement in the orchestrator's API server, CLI tools, or dashboards to perform privileged actions or read restricted data.

T-CO-DATA-DISC: Unauthorized disclosure of sensitive data via the container orchestrator

An attacker exploits vulnerabilities, excessive privileges, or misconfigured access controls in the orchestrator's API server, dashboard, or CLI to obtain sensitive data from managed workloads or control-plane resources.

This may include data in use (e.g. reading container memory or runtime secrets), data at rest (e.g. accessing container images, volumes, or configuration files), and data in transit (e.g. capturing pod-to-pod or service communications).

T-CO-AVAIL-DOS: DoS via resource exhaustion in the container orchestrator

An attacker overwhelms the orchestrator's control plane, APIs, or schedulers with excessive requests, malformed deployment manifests, or frequent scaling operations. This can lead to delays or complete unavailability in workload scheduling, management, and monitoring.

T-CO-LOG-INT: Insufficient logging or log tampering in the Container Orchestrator

An attacker targets the orchestrator's event and audit logging systems to suppress or alter records of critical cluster events, administrative actions, or API calls. This can allow malicious changes-such as unauthorized deployments, RBAC policy modifications, or workload scheduling-to go undetected.

T-CO-UPD-COMP: Malicious or compromised update in the Container Orchestrator

An attacker injects, tampers with, or replays updates to the orchestrator's control plane components (API server, scheduler, controller manager) or its plugins/extensions.

A compromised orchestrator update could allow the attacker to modify workload scheduling, bypass RBAC rules, or deploy malicious workloads across all managed nodes.

T-CO-CFG-COMP: Unauthorized or insecure configuration in the Container Orchestrator

An attacker exploits insecure default settings (such as anonymous API access, default administrator accounts, unrestricted inter-pod communication, or permissive admission controller policies) or performs unauthorized modifications of orchestrator configurations through its management interfaces (API server, dashboard, CLI) or configuration stores. Such weaknesses can compromise workload isolation, weaken network segmentation, bypass RBAC enforcement, or allow unverified workloads to run. Because the orchestrator governs multiple containers and services, insecure or unauthorized configurations can rapidly escalate into cluster-wide compromise or lateral movement between workloads. When orchestrator-level configuration is propagated to container engines or container runtimes, the configuration source and propagation path are authenticated and authorized to prevent unauthorized configuration changes from cascading across the environment.

T-CO-SC-COMP: Supply chain compromise in a Container Orchestrator

An attacker injects malicious code into a dependency (for example, a module, plugin, or API library) used in the orchestrator's control plane.

Because the orchestrator controls all cluster workloads and nodes, the compromise propagates to the entire environment.

B.2.3.2 Container Engine

T-CE-INTF-ESC: Engine-level cross-container interference

A malicious container exploits weaknesses in engine-level isolation enforcement (configuration, API exposure, or plugin vulnerabilities) to interfere with other containers running under the same engine.

T-CE-INTEG-COMP: Integrity compromise of container engine and managed images

An attacker tampers with the container engine's binaries, configuration files, or locally cached container images. The compromised engine can then execute manipulated workloads or disable controls, while still appearing healthy to the orchestrator.

T-CE-UNAUTH-ADMIN: Unauthorized access to container engine control endpoints

An attacker gains access to the container engine's local or remote control interfaces (e.g. Docker API socket, containerd gRPC API) due to weak authentication, unprotected sockets, or lack of TLS. This gives direct control over container lifecycle operations and engine configuration.

T-CE-UNAUTH-FUNC: Unauthorized privileged operations or data access in the container engine

An attacker with authenticated but restricted engine API access (local socket or remote endpoint) exploits weak authorization to invoke high-risk operations beyond their intended role.

T-CE-DATA-DISC: Unauthorized disclosure of sensitive data via the container engine

An attacker gains unauthorized access to container image layers, configuration files, or secrets stored locally on the engine host due to weak encryption, insufficient file permissions, or misconfigured access to control sockets.

This may include data in use (e.g. reading secrets from running container processes), data at rest (e.g. accessing local container image storage), and data in transit (e.g. intercepting traffic between containers using the engine's default network).

T-CE-AVAIL-DOS: DoS via resource exhaustion in the container engine

An attacker exploits the container engine's local API or socket to trigger excessive image pulls, container starts/stops, or logging operations, consuming CPU, memory, disk I/O, or network bandwidth on the host.

T-CE-LOG-INT: Insufficient logging or log tampering in the Container Engine

An attacker with access to the container engine API or control socket manipulates lifecycle logs (start/stop events, image pulls, error messages) to remove or alter records of their actions. Since the engine directly manages container execution, these logs are often key for local forensics.

T-CE-UPD-COMP: Malicious or compromised update in the Container Engine

An attacker injects or tampers with updates to the container engine binaries or API services. Since the engine directly interfaces with the host OS and launches containers, a compromised update can be used to execute arbitrary code with elevated privileges.

T-CE-CFG-COMP: Unauthorized or insecure configuration in the Container Engine

An attacker exploits insecure default settings (such as an exposed API socket without TLS/authentication, unrestricted default bridge networking, or permission to run privileged containers) or performs unauthorized modifications to container engine daemon or runtime configurations (e.g. Docker daemon JSON, container runtime parameters). Such weaknesses can bypass container isolation, allow unsafe container operations, or provide a direct path to escalate from container workloads to the host operating system. Because the container engine mediates between containers, the Container Runtime Stack (CRS), and the host OS, insecure or unauthorized configurations can undermine both workload security and host-level protections. If configuration changes originate from the orchestrator or external automation, the container engine only accepts configuration updates from authenticated and authorized sources to prevent unintended cluster-wide propagation of insecure settings.

T-CE-SC-COMP: Supply chain compromise in a Container Engine

A dependency used by the CE, such as a registry client library, image build tool, or network driver, is compromised during development or distribution.

This malicious code can be triggered when the CE pulls or builds images, or when it sets up container networking.

B.2.3.3 Container Runtime System

T-CRS-INTF-ESC: Runtime-level container breakout

A malicious container exploits flaws in OS-level isolation mechanisms or runtime implementation to escape its sandbox and interfere directly with the host or peer containers.

T-CRS-INTEG-COMP: Runtime-level tampering and unverified trust

An attacker modifies runtime binaries, configuration, or container state data (namespaces, cgroups, mounts) to disable or bypass security controls. Because the orchestrator and engine may not verify the runtime's state, a compromised runtime can operate undetected.

T-CRS-UNAUTH-ADMIN: Unauthorized access to container runtime control mechanisms

An attacker directly interacts with the container runtime binary or its control interfaces, bypassing higher-level engine or orchestrator authentication. This may occur if the runtime is invoked manually or through a compromised local account.

T-CRS-UNAUTH-FUNC: Unauthorized privileged operations or data access in the container runtime system

An attacker with authenticated local or delegated runtime access exploits insufficient authorization to manipulate running containers or modify runtime configurations directly.

T-CRS-DATA-DISC: Unauthorized disclosure of sensitive data via the container runtime system

An attacker exploits runtime-level weaknesses to read sensitive data from running containers or host resources that should be isolated.

This may include data in use (e.g. reading contents from container or host memory), data at rest (e.g. accessing mounted volumes or container filesystem layers), and data in transit (e.g. intercepting inter-container or container-host communications).

T-CRS-AVAIL-DOS: DoS via resource exhaustion in the container runtime system

An attacker running inside a container exploits runtime isolation weaknesses to monopolize CPU, memory, or I/O resources, or floods inter-container or container-to-host communication channels to degrade performance.

T-CRS-LOG-INT: Insufficient logging or log tampering in the Container Runtime System

An attacker operating inside a container exploits insufficient runtime logging or insecure log handling to conceal actions such as privilege escalation attempts, restricted syscalls, or file system access violations. The runtime's own log, if tampered with, may give a false picture of container activity.

T-CRS-UPD-COMP: Malicious or compromised update in the Container Runtime System

An attacker injects, tampers with, or replays updates to the low-level runtime responsible for creating and managing containers. Since the runtime enforces isolation (namespaces, cgroups, seccomp), a compromised update could remove or weaken these controls.

T-CRS-CFG-COMP: Unauthorized or insecure configuration in the Container Runtime System

An attacker exploits insecure defaults (such as lack of enforced seccomp/AppArmor profiles, missing cgroup limits for CPU, memory, or I/O, or permissive filesystem mounts that expose sensitive host paths) or introduces unauthorized changes to runtime-level parameters and configuration files. Such weaknesses undermine container isolation, disable syscall filtering, weaken resource constraints, or allow containers to access sensitive host resources. Because the container runtime system directly enforces workload isolation and resource governance, insecure or unauthorized configurations at this layer can enable container escapes, denial-of-service attacks, or privilege escalation against the host. Where runtime configuration is applied automatically from a container engine or orchestrator, the runtime ensures that such configuration inputs are authenticated and authorized before application, to prevent malicious or unintended configuration propagation.

T-CRS-SC-COMP: Supply chain compromise in a Container Runtime System

An attacker compromises a low-level runtime dependency, such as a system call handling library, sandboxing module, or container image unpacker, used by the CRS.

The malicious code executes whenever the CRS starts or runs containers, potentially enabling container escape.

B.2.4 Hardware-level common threat

T-ALL-HW-ACCESS: Compromise via hardware-level or platform trust components

An attacker with access to hardware-level capabilities (for example compromised CPU, firmware, hardware security modules, or physical access to deployed systems) may compromise the hypervisor, container runtime system, or associated platform components below the software control plane. Such attacks may exploit manufacturer backdoors in processors or firmware, supply chain implants in platform components, or physical tampering vectors.

Because hardware-level compromise can operate at privilege levels that exceed the visibility and control of software-based protection mechanisms, these attacks may bypass or undermine boot integrity verification, runtime integrity checks, attestation, and isolation mechanisms defined in the present document.

The present document does not require products to fully prevent all forms of hardware-level compromise. Instead, for requirements that rely on platform-anchored, hardware-assisted, or externally protected trust mechanisms, manufacturers are expected to:

- use available platform trust anchors and protection mechanisms where appropriate;
- provide software-based or hybrid protections where feasible to achieve equivalent security outcomes; and
- document trust assumptions and residual risks associated with hardware components, including the potential impact of compromise consistent with T-ALL-HW-ACCESS.

B.3 Risk Assessment Process

The risk assessment process focuses on evaluating the product within its defined use case and deployment environment, following a logical sequence of steps:

1) Define the product and its context: Identify the product being assessed, its intended use case, and the specific deployment environment it is intended for. This provides the foundational context for the assessment.

2) Identify product threats: Identify potential cybersecurity threats that could exploit vulnerabilities.

For each identified threat, the following steps should be performed:

3) Analyse risk factors for likelihood and impact as defined in clause B.5: This step determines the product's cybersecurity risk by evaluating its characteristics related to potential threats. For each identified threat, and for every relevant risk factor, the following guiding question should be applied: *"How does this factor influence the likelihood or impact if the threat were to occur?"*. Each threat is evaluated against a defined set of risk factors covering both Likelihood and Impact. These factors are scored on a three-level scale (Low = 1, Medium = 2, High = 3) using the definitions provided in clause B.4.

The analysis involves the following sub-steps:

- Identifying key risk factors: Determine the main aspects of the product's operation within that environment that influence its cybersecurity risk (e.g. connectivity, administrative context, configuration volatility, impact of compromise, integration with critical systems). These include factors that affect either:

    1) the likelihood of threat exploitation; or

    2) the severity of consequences if the threat materializes.

These factors serve as proxies for the likelihood of a threat materializing and the magnitude of its impact.

- Defining scoring: Assign numerical scores (1 for low risk, 2 for medium, 3 for high) to each risk factor based on how it influences the likelihood of a threat event and the magnitude of its potential impact.

4) Derive overall likelihood score: Sum the scores of all risk factors identified as influencing "Likelihood". Map this total score to the Likelihood level (1-3) using the thresholds defined in clause B.5.2.

5) Derive overall impact score: Sum the scores of all risk factors identified as influencing "Impact". Map this total score to the Impact level (1-3) using the thresholds defined in clause B.5.3.

6) Determine final risk level: Using the derived overall Likelihood score and overall Impact score, consult the risk matrix (clause B.6) to determine the final, overall risk level (Low, Medium, High) for the specific use case.

7) Document and review: Record all assessment findings, including the rationale for scores and the determined risk level.

B.4 Risk Factor Scoring

Table B.4-1 provides the detailed scoring rubric to be used for assessing cybersecurity risks of products within their use cases and deployment environments. Each risk factor is explicitly linked to whether it primarily influences likelihood or impact. The 1-3 ratings of the risk factors are summed per type (Likelihood or Impact) to derive the overall likelihood and impact scores.

Table B.4-1: Risk factors

Tag	Factor	Type	Description	Low (1)	Medium (2)	High (3)
EXT	External Network Exposure	Likelihood	Measures how reachable the VES/CES is from untrusted networks, including public internet and cross-tenant networks. Higher exposure increases the probability of remote exploitation.	Internal-only access; management on isolated VLAN/bastion; strict firewall segmentation.	Limited/controlled external access (VPN, allow-list, jump host).	Public internet-facing endpoints or broad cross-tenant network paths.
TEN	Tenant Multiplicity & Trust	Likelihood	Captures the number of tenants and their trust level. Multi-tenancy with untrusted tenants increases the chance of malicious insiders or cross-tenant attacks.	Single trusted tenant.	Few semi-trusted tenants.	Many untrusted tenants (public/multi-organization).
CTL	Control-Plane Exposure, Access Control & Detectability	Likelihood	Evaluates the exposure, strength of authentication/authorization on management interfaces, and the extent to which malicious actions on the control plane would be detectable.	Admin access isolated, MFA + RBAC enforced, centralized logging alerts on anomalous admin activity.	Internal network access, standard RBAC, partial monitoring or periodic audits.	Externally reachable control plane, weak RBAC, or limited/no monitoring of administrative actions.
SUP	Update & Supply-Chain Assurance	Likelihood	Assesses the security of update mechanisms, software/image sources, and supply-chain validation. Weak assurance can allow malicious updates or images.	Signed/verified only; private mirrors; staged/canary updates.	Vendor-signed updates/images; occasional direct pulls.	Direct pulls from public registries/vendor repositories; unsigned allowed.
PRV	Attack Complexity & Privileged Footprint	Likelihood	Represents the attacker skill/resources required to exploit the privileged software surface. Higher privileged footprint or weaker hardening → lower attacker skill required.	Minimal attack surface, compiler + runtime hardening (ASLR, SMEP/SMAP/CET), requires advanced attacker capability.	Mixed hardening, moderate privileged surface, requires skilled attacker with tools.	Large privileged surface, device pass-through, weak/no hardening - exploitable by unsophisticated attackers.
CHG	Change/Workload Volatility	Impact	Evaluates how frequently the configuration and workloads change. Higher volatility increases the likelihood of misconfigurations or unpatched instances.	Static, rarely changes.	Periodic changes (weekly/monthly).	Highly dynamic (daily/hourly), autoscaling, frequent reconfiguration.
ISO	Isolation Breakout Consequence	Impact	Assesses the damage if the VM/container isolation boundary is broken. Higher values mean more severe lateral movement or system compromise.	No sensitive assets exposed; minimal lateral movement possible.	Exposure of internal systems/data; contained blast radius.	Cross-tenant or critical system compromise; admin plane reachable.

Tag	Factor	Type	Description	Low (1)	Medium (2)	High (3)
DAT	Data/Safety/Criticality	Impact	Measures the sensitivity of the data and any safety or regulatory implications of compromise. Higher scores indicate severe operational, safety, or compliance impacts.	Public or low-sensitivity data; no safety impact.	Confidential/business data; moderate operational disruption.	Regulated/Personal data/financial/eHealth data; safety hazard or critical infrastructure outage.
ORB	Orchestration Dependency & Blast Radius	Impact	Evaluates how much damage an orchestrator compromise could cause. Higher values reflect large-scale or critical workload control.	Stand-alone host; no orchestrator or read-only integration.	Basic orchestrator; tenant- or namespace-scoped control.	Full control of critical workloads across multi-node clusters/sites.
DRC	Detection & Recovery Capability	Impact	Measures the ability to detect, contain, and recover from a compromise. Poor detection/recovery increases the residual impact of an incident.	Centralized logging/alerting; immutable backups; tested recovery plans.	Adequate logging/backups; periodic recovery tests.	Minimal logging/alerting; weak or untested recovery procedures.

B.5 Calculation and Mapping to Likelihood/Impact Levels

B.5.1 General

This clause defines how the total scores from Table B.4-1 in clause B.4 are translated into the 1-3 Likelihood and Impact levels used in the risk matrix (clause B.6).

B.5.2 Likelihood Score

Table B.4-1 in clause B.4 defines five Likelihood factors:

- EXT (External Network Exposure)
- TEN (Tenant Multiplicity & Trust)
- CTL (Control-Plane Exposure, Access Control & Detectability)
- SUP (Update & Supply-Chain Assurance)
- PRV (Privileged Footprint & Hardening)

Each factor is scored from 1 (Low) to 3 (High):

- Maximum possible Likelihood score: 5 factors × 3 = 15
- Minimum possible Likelihood score: 5 factors × 1 = 5

The resulting total score is mapped to three levels as follows:

Table B.5.2-1: Likelihood score

Total Likelihood Score Range	Likelihood Level	Description
5 - 8	Low (1)	The event is highly unlikely to occur under normal circumstances.
9 - 12	Medium (2)	The event may occur occasionally in enterprise or cloud deployments.
13 - 15	High (3)	The event is expected to occur or has a high probability of occurring.

B.5.3 Impact Score

Table B.4-1 in clause B.4 defines five Impact factors:

- CHG (Change/Workload Volatility)
- ISO (Isolation Breakout Consequence)
- DAT (Data/Safety/Criticality)
- ORB (Orchestration Dependency & Blast Radius)
- DRC (Detection & Recovery Capability)

Each factor is scored from 1 (Low) to 3 (High).

- Maximum possible Impact score: 5 factors × 3 = 15
- Minimum possible Impact score: 5 factors × 1 = 5

The resulting total score is mapped to three levels as follows:

Table B.5.3-1: Impact score

Total Impact Score Range	Impact Level	Description
5 - 8	Low (1)	Minimal disruption. No significant loss of confidentiality, integrity, or availability. No impact on safety or critical infrastructure.
9 - 12	Medium (2)	Noticeable disruption requiring recovery. Moderate impact on confidentiality, integrity, or availability. Potential for indirect impact on safety or critical infrastructure.
13 - 15	High (3)	Severe disruption, service interruption, or data loss. Major impact on confidentiality, integrity, or availability. May directly affect safety or critical infrastructure.

B.6 Risk Matrix

The final overall risk level is determined by combining the calculated likelihood level (clause B.5.2) and impact level (clause B.5.3) using the matrix below:

Table B.6-1: Risk matrix

Impact \ Likelihood	1 = Low	2 = Medium	3 = High
3 = High	Medium	High	High
2 = Medium	Low	Medium	High
1 = Low	Low	Low	Medium

B.7 Risk Methodology Applied to Use Cases

B.7.1 General

This clause illustrates the application of the methodology to products deployed within the VES use cases (UC-V1, UC-V2, UC-V3) and CES use cases (UC-C1, UC-C2, UC-C3). The use cases are defined in clause 4.6, while the corresponding threats applicable to these products are described in clause B.2.

B.7.2 Risk Evaluation of VES use cases

B.7.2.1 Hypervisor Type I

Table B.7.2.1-1: Hypervisor Type I - Risk evaluation of VES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP1-INTF-ESC: Inter-VM interference / VM escape														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	In UC-V1 the Type-1 hypervisor runs on a dedicated, isolated host with a very small set of non-critical VMs, limited external connectivity (EXT=1, TEN=1) and strictly local administration. Even if a VM escape occurs, lateral movement remains confined to that single host and low-value workloads, so both likelihood and impact are assessed as Low.
V2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	In UC-V2, the Type-1 hypervisor is part of an enterprise cluster with more VMs per tenant, shared management networks and standard hardening (EXT=1, TEN=2, CTL=3). A successful VM escape would expose all VMs owned by that enterprise tenant on that cluster, making both the feasibility and the consequences noticeably higher than in UC-V1.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-V3, Type-1 hypervisors underpin multi-tenant critical clouds where untrusted tenants co-exist and regulated or safety-relevant workloads are hosted (TEN=3, DAT=3, ORB=3). A VM escape here enables cross-tenant compromise and direct access to high-criticality data, so both likelihood and impact are rated High.
T-HYP1-INTEG-COMP: Integrity compromise of workloads / hypervisor components														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	For UC-V1, tampering with hypervisor binaries or VM images typically requires local access to a single host; workloads are static and non-critical, and orchestration blast radius is limited (ORB=1). As a result, integrity compromise remains a localized issue with Low impact on operations.
V2	1	2	3	2	2	2	2	2	2	3	10 → Med	11 → Med	Medium	In UC-V2, hypervisor binaries, kernel modules and VM images are centrally managed for business workloads. If an attacker tampers with these components on an enterprise VES, they can gain persistent control over all VMs of that tenant (ISO=2, ORB=2) and manipulate business services, so both likelihood and impact are Medium.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V3	3	3	3	3	3	3	2	3	3	3	15 → High	14 → High	High	In UC-V3, a compromise of Type-1 hypervisor components (e.g. signed modules, microcode integrations) affects multi-tenant, high-assurance environments. Any persistent backdoor or binary tampering undermines the entire trust anchor for critical workloads across tenants (ISO=2 but DAT=3, ORB=3), justifying High likelihood and High impact.
T-HYP1-UNAUTH-ADMIN: Unauthorized access to management interfaces														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	In UC-V1, direct admin interfaces of the Type-1 hypervisor are usually reachable only from an isolated management network or via local console, and the host runs non-critical VMs. Even if authentication is weak, the lack of broad network exposure and the low value of the workloads keep both likelihood and impact Low.
V2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	For UC-V2, the Type-1 management plane (e.g. vSphere, vendor APIs) is reachable from the enterprise network and is a high-value target (EXT=1, CTL=3). A compromise of admin credentials or interfaces gives full control over all enterprise VMs, but the effects are still contained within that tenant/domain, leading to Medium risk.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-V3, Type-1 management APIs, even if segregated, are critical control points for multi-tenant and regulated workloads. Unauthorized access to these interfaces would allow massive cross-tenant operations, uncontrolled VM management, and potential breach of critical systems, warranting High likelihood and High impact.
T-HYP1-UNAUTH-FUNC: Unauthorized privileged actions														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	Low	Within UC-V1, a misconfigured RBAC or over-permissive role on the Type-1 hypervisor can allow privilege abuse (e.g. VM snapshots, power-off) but only on a few local, non-critical VMs. Given strong isolation from external networks and the limited blast radius, both likelihood and impact remain Low.
V2	1	2	3	1	3	2	2	2	2	2	10 → Med	10 → Med	Medium	In UC-V2, enterprise deployments may assign operator roles to multiple administrators or automation tools. If authorization checks are weak (CTL=3, PRV=3), a compromised low-privilege account can perform administrative actions (e.g. attach disks, modify vSwitches) across the tenant's VMs, making both likelihood and impact Medium.
V3	2	3	2	3	3	3	3	3	3	3	13 → High	15 → High	High	In UC-V3, weak or inconsistent authorization on the Type-1 management plane allows a compromised operator account to perform privileged actions affecting many untrusted tenants (TEN=3, ORB=3) and high-criticality workloads. This drives both likelihood and impact to High.
T-HYP1-DATA-DISC: Data disclosure														
V1	1	1	2	1	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-V1, Type-1 hypervisors host non-critical workloads with low-sensitivity data (DAT=1) and minimal connectivity. Even if hypervisor misconfigurations or weak encryption expose VM disks or inter-VM traffic, the resulting confidentiality impact is limited, so both likelihood and impact are Low.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	For UC-V2, the Type-1 hypervisor stores and processes business data and may support backups or snapshots. Misconfigurations or vulnerabilities leading to access to VM images or memory dumps would significantly affect the enterprise tenant, but not multiple independent tenants, leading to Medium risk.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, Type-1 hypervisors underpin clouds handling regulated and critical workloads (e.g. eHealth, finance, telecom core). A breakout that accesses VM RAM, disks, or east-west traffic can disclose highly sensitive multi-tenant data (DAT=3, ISO=3, ORB=3), resulting in High likelihood and High impact.
T-HYP1-AVAIL-DOS: DoS via resource exhaustion														
V1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-V1, resource exhaustion attacks generally require local or tightly scoped access, target a small number of VMs, and affect functions that are not business- or safety-critical. Even if CPU, memory or I/O are saturated, only a single host and low-value workloads are impacted, so likelihood and impact are Low.
V2	1	2	2	2	2	2	2	2	2	2	9 → Med	10 → Med	Medium	For UC-V2, noisy-neighbour effects or malicious workloads can cause CPU steal, memory pressure, or shared-I/O bottlenecks on Type-1 hosts that carry many enterprise workloads. While blast radius is still tenant-scoped, such events can cause notable service disruption (CHG=2, ORB=2), driving likelihood and impact to Medium.
V3	2	3	3	2	3	3	3	2	3	2	13 → High	13 → High	High	In UC-V3, shared Type-1 hypervisors run heterogeneous, often untrusted tenants at high density. Misconfigured overcommitment or deliberate resource exhaustion can create systemic availability issues across many critical services (TEN=3, CHG=3, ORB=3). Therefore, both likelihood and impact are assessed as High.
T-HYP1-LOG-INT: Insufficient logging / log tampering														
V1	1	1	1	1	1	1	1	1	1	2	5 → Low	6 → Low	Low	In UC-V1, logging on the Type-1 hypervisor may be minimal, and local administrators or malware could clear or modify logs. However, the environment has low data sensitivity and limited orchestration (DAT=1, ORB=1), so the inability to trace events has a Low overall impact.
V2	1	2	2	2	2	2	1	2	2	2	9 → Med	9 → Med	Medium	In UC-V2, insufficient audit trails on hypervisor admin actions, VM lifecycle events, and configuration changes delay detection of targeted attacks on business workloads. Enterprise security monitoring exists but may not fully cover hypervisor logs (DRC=2), so both likelihood and impact are Medium.
V3	2	3	3	2	3	2	2	3	3	3	13 → Med	13 → High	High	In UC-V3, gaps or tampering in Type-1 logging and audit mechanisms seriously undermine forensic readiness and incident containment. In multi-tenant critical clouds, missing or untrustworthy logs mean that cross-tenant attacks may remain undetected and unbounded in time, resulting in High likelihood but High impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP1-UPD-COMP: Malicious or compromised update														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, updates to the Type-1 hypervisor may be done manually or infrequently, sometimes from offline media or vendor repositories. A compromised image would affect a single host with non-critical workloads, and some manual verification is often possible, so both likelihood and impact stay Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	Medium	For UC-V2, centralized yet sometimes ad-hoc update procedures (SUP=2) mean that delayed or inconsistent patching can allow malicious or tampered hypervisor updates to be applied across an enterprise cluster. This can compromise all VMs owned by that tenant, giving Medium likelihood and impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, Type-1 hypervisors rely on highly automated, large-scale update pipelines. A compromise of the update supply chain or signing infrastructure would propagate to many critical hosts simultaneously (SUP=3, ORB=3), making both likelihood and impact High.
T-HYP1-CFG-COMP: Insecure or Unauthorized configuration														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-V1, misconfigurations (e.g. leaving a debug interface enabled or using permissive vSwitch settings) are usually applied locally and affect a small, static VM set. The environment is not exposed to untrusted tenants, so while isolation can be weakened, the overall risk remains Low.
V2	2	2	3	2	2	2	2	2	2	2	11 → Med	10 → Med	Medium	For UC-V2, centralized management of many VMs introduces more frequent changes and complex configuration (CHG=2). Errors such as exposing management services, disabling security features or misconfiguring virtual networks can expose sensitive enterprise workloads, leading to Medium likelihood and Medium impact.
V3	2	3	3	2	3	3	3	3	3	3	13 → High	15 → High	High	In UC-V3, configuration baselines have to enforce strong isolation, least privilege, and hardened defaults. Insecure or unauthorized changes (e.g. tenant-visible management networks, overly permissive resource mappings) can immediately affect multiple critical tenants and services, so both likelihood and impact are High.
T-HYP1-SC-COMP: Supply chain compromise														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, a compromised driver, library, or firmware component used by the Type-1 hypervisor yields a backdoor on one isolated host. Since workloads are low-value and connectivity is limited, the damage is confined, so likelihood and impact are Low.
V2	2	2	2	3	2	2	2	2	2	2	11 → Med	10 → Med	Medium	In UC-V2, the Type-1 stack depends on a broader supply chain (Virtualisation platform, OS components, drivers). A poisoned component can introduce persistent backdoors affecting all enterprise VMs on the cluster, but still within one tenant domain, resulting in Medium likelihood and Medium impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, supply-chain compromise of hypervisor components (e.g. low-level drivers or cryptographic libraries) scales across many tenants and critical services. Every VM on the affected hosts inherits the compromised code path, giving the attacker a privileged foothold with High likelihood and High impact.

B.7.2.2 Hypervisor Type II

Table B.7.2.2-1: Hypervisor Type II - Risk evaluation of VES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP2-HOST-ABUSE: Guest-to-host escalation														
V1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, a Type-2 hypervisor runs on a single host OS with few non-critical VMs and limited services exposed. An attacker in a guest can theoretically abuse host APIs or shared services, but low exposure and low data value keep both likelihood and impact Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, the host OS provides shared services (e.g. file shares, clipboard, device redirection) to multiple enterprise VMs. Weak isolation or host hardening allows a malicious VM to escalate into the host OS, from which it can access all internal workloads on that machine, giving Medium likelihood and impact.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, hosted Virtualisation may underpin critical or multi-tenant workloads; a successful guest-to-host escalation immediately compromises the host OS and all VMs it controls (ISO=3, DAT=3, ORB=3). Given the large attack surface of general-purpose OSes, both likelihood and impact are High.
T-HYP2-INTEG-COMP: Integrity compromise of hosted hypervisor														
V1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	For UC-V1, tampering with the Type-2 hypervisor binary or VM images on the host filesystem typically requires local access and affects only non-critical workloads. The blast radius is limited to one host, so likelihood and impact are Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, compromise of host OS or hypervisor application binaries directly affects the integrity of all VMs on that host and may propagate through shared images or templates within the enterprise. However, the effect is still scoped to one internal tenant environment, leading to Medium risk.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-V3, hosted Virtualisation is used for high-assurance workloads; any manipulation of binaries, modules or VM images via the host OS or packaging system undermines trust in multiple critical services. This justifies High likelihood and High impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP2-UNAUTH-ADMIN: Unauthorized admin access														
V1	1	1	2	1	1	1	1	1	1	1	6 → Low	5 → Low	Low	In UC-V1, the host OS admin account and the Type-2 hypervisor console are typically accessible only locally or on a restricted network, and control non-critical VMs. Even if passwords are weak, compromise remains local to a single device, making both likelihood and impact Low.
V2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	For UC-V2, host OS and hypervisor admin interfaces are reachable from the enterprise network and control multiple business VMs (CTL=3). Unauthorized access to these accounts enables full control over internal workloads, but still within one enterprise boundary, so the risk is Medium.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, internet-reachable or widely exposed host OS admin accounts and Type-2 management consoles represent a systemic control point. Their compromise yields administrative control over critical or regulated workloads and potentially multiple tenants, so both likelihood and impact are High.
T-HYP2-UNAUTH-FUNC: Unauthorized privileged actions														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-V1, misconfigured permissions in the Type-2 application or host OS may allow local users to perform limited privileged actions, but the small number of non-critical VMs and limited connectivity keep the overall risk Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, enterprise deployments often have multiple operators or automation agents interacting with the hypervisor application. Weak separation of roles or coarse-grained permissions allow unauthorized privileged actions that affect many internal workloads (PRV=2, ISO=2), giving Medium likelihood and impact.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, insufficient authorization controls in the Type-2 hypervisor or host OS allow a compromised user or process to perform high-impact operations (e.g. mount host paths, change networking, access snapshots) across critical tenants. This drives both likelihood and impact to High.
T-HYP2-DATA-DISC: Sensitive data disclosure														
V1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-V1, VMs on a hosted hypervisor process low-sensitivity data and are often isolated physically or logically. Even if an attacker reads VM image files or host-mediated traffic, the confidentiality impact is minimal, resulting in Low likelihood and Low impact.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, a Type-2 hypervisor can host enterprise workloads whose VM image files, snapshots, or memory may contain confidential business data. Exploiting host OS weaknesses or hypervisor flaws to access these artefacts affects the enterprise tenant significantly, but not other independent tenants, leading to Medium risk.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-V3, Type-2 environments may handle highly sensitive or regulated data on general-purpose OSes. If an attacker gains access to VM images, RAM, or inter-VM traffic via the host, multiple

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
														critical services and tenants can be affected, so both likelihood and impact are High.
T-HYP2-AVAIL-DOS: Resource exhaustion														
V1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-V1, a DoS caused by a single VM or host-level process consuming CPU, memory, or I/O affects only a standalone host with non-critical workloads, yielding Low likelihood and impact.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	For UC-V2, resource exhaustion on a host OS running a Type-2 hypervisor can disrupt many internal business VMs (e.g. via runaway processes or misconfigured resource limits). However, the effect is still constrained to one enterprise environment, so likelihood and impact are Medium.
V3	3	3	2	2	3	3	3	3	3	2	13 → High	14 → High	High	In UC-V3, hosted Virtualisation may carry critical services or multi-tenant workloads; noisy-neighbour effects or deliberate resource abuse at host level can degrade or interrupt many high-criticality services simultaneously (CHG=3, ORB=3). This justifies High likelihood and High impact.
T-HYP2-LOG-INT: Logging weakness/tampering														
V1	1	1	1	1	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-V1, logs for the Type-2 hypervisor and host OS are often stored locally and may not be centrally monitored. While this allows some tampering, the low-sensitivity nature of the workloads limits the overall impact, so risk is Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, weak or inconsistent logging of VM lifecycle events, admin actions, or host OS changes makes enterprise incident detection slower. Attackers can persist longer within internal workloads (DRC=2), which supports Medium likelihood and impact.
V3	2	3	3	2	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, insufficient or tamperable logs on the host OS and Type-2 hypervisor significantly weaken forensic and recovery capabilities across critical workloads. Multi-tenant or regulated services then lack reliable evidence of compromise, resulting in High likelihood and High impact.
T-HYP2-UPD-COMP: Malicious/compromised update														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, updates to the hosted hypervisor application are often done manually or automatically by local admins. A malicious installer can compromise that single host and its few non-critical VMs, but the limited scale and occasional update frequency make the overall risk Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, admins may download and install Type-2 hypervisor updates directly from vendors or repositories on multiple enterprise hosts. A poisoned installer or downgraded package can compromise many internal workloads, so both likelihood and impact are Medium.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, automated or large-scale update processes for hosted hypervisors and host OS components mean a single compromised package or repository can propagate quickly to many critical nodes (SUP=3, ORB=3). This supports High likelihood and High impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-HYP2-CFG-COMP: Insecure or Unauthorized configuration														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-V1, insecure defaults (e.g. permissive host-guest shared folders, bridged networking) or local misconfigurations affect only a small number of non-critical VMs. The lack of multi-tenancy and external exposure keeps likelihood and impact Low.
V2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	In UC-V2, Type-2 hypervisors on enterprise hosts may be configured with convenient but insecure settings (e.g. broad host path mappings, weak isolation between lab and production VMs). These configurations enable lateral movement and unauthorized access between internal workloads, giving Medium risk.
V3	2	3	3	2	3	3	3	3	3	3	14 → High	14 → High	High	In UC-V3, insecure Type-2 configurations (such as open shared folders, permissive host networking, or control channels without authentication) directly break isolation between high-assurance tenants and critical services. Given the strong reliance on configuration correctness, both likelihood and impact are High.
T-HYP2-SC-COMP: Supply chain compromise														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-V1, a malicious driver or library used by the Type-2 hypervisor executes only on a single, isolated host. While it provides a backdoor to that system, the limited connectivity and low-value workloads constrain the impact, so likelihood and impact are Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-V2, enterprises may deploy Type-2 hypervisors at scale using OS packages or vendor installers. A compromised dependency (e.g. kernel module, library) can introduce persistent, hard-to-detect backdoors on multiple hosts and affect many internal workloads, leading to Medium likelihood and impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-V3, Type-2 hypervisors and host OS components depend on broader upstream ecosystems. A supply-chain attack on drivers, libraries, or packaged modules can propagate widely, compromising host OS and hypervisor instances across critical tenants, thus supporting High likelihood and High impact.

B.7.2.3 M&O System

Table B.7.2.3-1: M&O System – Risk evaluation of VES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-M&O-CTRL-PLANE: Control-plane abuse leading to cross-workload interference and privilege escalation														
V1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-V1, the M&O system manages a single isolated hypervisor host with few non-critical VMs and limited administrative network exposure. Control-plane access is restricted to local or isolated management networks (CTL=2, EXT=1), and the blast radius of any control-plane abuse is limited to that single host environment (ORB=1). Even if an attacker manipulates scheduling or networking policies, the impact remains confined to low-value workloads on a single node, so both likelihood and impact are assessed as Low.
V2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-V2, the M&O system orchestrates hypervisors across an enterprise cluster, with management interfaces exposed on the enterprise network (CTL=3, EXT=1). An attacker who gains control-plane access can manipulate scheduling, networking, or storage policies to reach workloads across that tenant's environment (ISO=2, ORB=2). While the blast radius is still limited to one organizational tenant, the ability to affect many VMs and business services makes both likelihood and impact Medium.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	**High**	In UC-V3, the M&O system governs multi-tenant, highly dynamic hypervisor clusters processing critical or regulated workloads (TEN=3, DAT=3). Control-plane interfaces may be externally reachable or exposed to many administrators and automation pipelines (EXT=3, CTL=3), and a successful control-plane abuse can cascade scheduling changes, network policy overrides, or resource reallocations across the entire multi-tenant fleet (ORB=3, ISO=3). Both likelihood and impact are therefore High.
T-M&O-INTEG-TRUST: Misplaced trust in compromised platform														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	**Low**	In UC-V1, the M&O system manages a single static host with limited attestation requirements. If the hypervisor host is compromised and feeds falsified state data, the impact remains limited to a single low-value environment. The combination of minimal orchestration blast radius (ORB=1) and low data sensitivity (DAT=1) keeps both likelihood and impact Low.
V2	1	2	3	2	2	2	2	2	2	3	10 → Med	11 → Med	**Medium**	In UC-V2, an enterprise M&O system may schedule workloads across multiple internal hypervisor hosts without rigorous per-host attestation or integrity verification. A compromised host that sends falsified health or state data can influence scheduling decisions, potentially exposing enterprise data across that tenant's infrastructure (ORB=2, ISO=2). Moderate enterprise monitoring may partially compensate (DRC=3). Both likelihood and impact are Medium.
V3	3	3	3	3	3	3	2	3	3	3	15 → High	14 → High	**High**	In UC-V3, high-assurance orchestration relies on trusted attestation of each hypervisor host in the fleet. If attestation mechanisms are absent or can be spoofed by a compromised host, the M&O may continue scheduling regulated or critical multi-tenant workloads onto a tampered host (TEN=3, DAT=3, ORB=3). The attacker gains a privileged position over all workloads placed on

														that host without triggering orchestrator-level alerts, giving High likelihood and High impact.
T-M&O-UNAUTH-ADMIN: Unauthorized access to orchestration/management plane														
V1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	**Low**	In UC-V1, M&O management interfaces are reachable only via local or isolated management networks, and they control a small number of non-critical VMs on a single host (EXT=1, TEN=1). Even if authentication is weak, the lack of external exposure and the minimal value of managed workloads keep both likelihood and impact Low.
V2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-V2, M&O management APIs and dashboards are reachable from the enterprise network and govern all hypervisors in the cluster (CTL=3, ORB=2). Unauthorized access via compromised credentials or missing MFA gives an attacker control over all enterprise VMs and hosts managed by that orchestration system. The effect is scoped to one organizational tenant but can impact many internal workloads, giving Medium likelihood and Medium impact.
V3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	**High**	In UC-V3, M&O administrative interfaces provide cluster-wide control over critical multi-tenant hypervisor fleets (TEN=3, ORB=3). External or broadly reachable control planes (EXT=3) combined with potentially weak authentication or compromised admin credentials represent the highest-value attack target in the environment. A successful breach gives the attacker unrestricted control over all hosted workloads and regulated data, making both likelihood and impact High.
T-M&O-UNAUTH-FUNC: Unauthorized privileged actions or data access in orchestration/management systems														
V1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-V1, M&O users or automation tools have limited roles managing a small set of non-critical VMs. Even if RBAC is misconfigured, the privilege abuse can only affect that single host's workloads, and the environment is not exposed to untrusted users, so both likelihood and impact are Low.
V2	1	2	3	1	3	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-V2, enterprise M&O platforms often delegate limited API access to operators, automation pipelines, and CI/CD systems. Insufficient RBAC granularity (CTL=3, PRV=3) can allow a compromised low-privilege account to perform cluster-level privileged operations (e.g. snapshot VMs of other users, reassign storage, modify network policies) affecting all VMs within the enterprise tenant (ORB=2), giving Medium likelihood and impact.
V3	2	3	2	3	3	3	3	3	3	3	13 → High	15 → High	**High**	In UC-V3, the M&O platform grants broad privileged access to automation systems, multi-team operators, and federated administrators managing critical or regulated workloads. Weak or inconsistent authorization enforcement allows over-privileged accounts to execute high-impact operations across many tenants (TEN=3, ORB=3), such as modifying isolation policies, migrating sensitive VMs to untrusted hosts, or accessing restricted data. This drives both likelihood and impact to High.
T-M&O-DATA-DISC: Unauthorized disclosure of sensitive data via Virtualisation orchestration/management system														
V1	1	1	1	1	1	1	1	1	1	2	5 → Low	6 → Low	**Low**	In UC-V1, M&O platforms manage a small number of non-critical VMs on an isolated host. Configuration data, VM snapshots, or orchestrated workload metadata have low sensitivity (DAT=1), and the management plane is not externally exposed (EXT=1). Even if access controls are weak, the resulting confidentiality impact is minimal, so both likelihood and impact are Low.

V2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-V2, the M&O system aggregates sensitive information about enterprise workloads - including VM configurations, network topology, API tokens, and orchestration secrets. Vulnerabilities or access-control weaknesses in the management API or dashboard can expose this data to internal adversaries or compromised accounts, affecting the entire enterprise tenant. The scope is tenant-limited but still significantly impacts business operations, giving Medium likelihood and impact.
V3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	**High**	In UC-V3, M&O platforms handle multi-tenant workload configurations, credentials, network overlays, and potentially memory dumps or runtime secrets for regulated or safety-critical services (DAT=3). A breach of M&O data stores or API outputs gives attackers a comprehensive map of the entire infrastructure alongside direct access to sensitive secrets (ISO=3, ORB=3). This supports High likelihood and High impact.
T-M&O-AVAIL-DOS: DoS via resource exhaustion in Virtualisation orchestration/management systems														
V1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	**Low**	In UC-V1, the M&O manages a single host with a small number of non-critical VMs. Flooding the management API or control loop with excessive requests would at most delay provisioning or management operations on that single node. Given the limited orchestration scope and non-critical workloads, both likelihood and impact are Low.
V2	1	2	2	2	2	2	2	2	2	2	9 → Med	10 → Med	**Medium**	In UC-V2, M&O schedulers and APIs handle ongoing lifecycle operations for multiple enterprise VMs. Excessive scheduling or provisioning requests - whether from a compromised VM or a misconfigured automation tool - can render management operations unresponsive and prevent timely maintenance, patching, or incident response across the enterprise tenant. Both likelihood and impact are Medium.
V3	2	3	3	2	3	3	3	2	3	2	13 → High	13 → High	**High**	In UC-V3, M&O control planes govern critical multi-tenant hypervisor fleets handling regulated workloads (TEN=3, ORB=3). An attacker who floods management APIs, generates excessive VM lifecycle events, or exploits autoscaling policies can paralyse the orchestration layer, preventing timely workload migration, failover, or maintenance across the entire cluster. This can directly degrade service availability for critical and regulated services, resulting in High likelihood and High impact.
T-M&O-LOG-INT: Insufficient logging or log tampering in Virtualisation orchestration/management systems														
V1	1	1	1	1	1	1	1	1	1	2	5 → Low	6 → Low	**Low**	In UC-V1, logging of M&O administrative events and VM lifecycle operations may be minimal or stored locally. Even if logs are incomplete or tampered, the low data sensitivity and minimal orchestration scope (DAT=1, ORB=1) limit the forensic impact, so overall risk remains Low.
V2	1	2	2	2	2	2	1	2	2	2	9 → Med	9 → Med	**Medium**	In UC-V2, M&O platforms are the primary source of lifecycle and administrative audit records for enterprise hypervisor clusters. Insufficient audit trails or ease of log modification at the M&O layer can delay detection of targeted configuration changes, unauthorized VM deployments, or privilege abuse within the enterprise environment. This warrants Medium likelihood and Medium impact.
V3	2	3	3	2	3	2	2	3	3	3	13 → High	13 → High	**High**	In UC-V3, the M&O system is the central source of audit truth for critical multi-tenant infrastructure. Tampering with or suppressing M&O logs eliminates

														evidence of cross-tenant access violations, unauthorized configuration changes, or privilege escalations affecting regulated workloads (DAT=3, ORB=3). The inability to reconstruct attack timelines also hampers breach containment and regulatory reporting, resulting in High likelihood and High impact.
T-M&O-UPD-COMP: Malicious or compromised update in Virtualisation orchestration/management systems														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-V1, M&O software updates are typically performed manually or infrequently and affect only a single host managing non-critical workloads. A malicious update could compromise the orchestration function on that node, but the limited scale and low workload sensitivity keep both likelihood and impact Low.
V2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-V2, enterprise M&O platforms may be updated via automated or semi-automated pipelines. A compromised update package (e.g. a tampered orchestration agent, plugin, or management dashboard) can be rolled out across the enterprise hypervisor cluster (SUP=2, ORB=2), giving an attacker persistent access to all managed VMs. This supports Medium likelihood and impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	**High**	In UC-V3, the M&O platform is managed at scale with highly automated update pipelines (SUP=3). A compromised update to the orchestration control plane propagates immediately to all connected hypervisors and workloads across critical, multi-tenant deployments (ORB=3, ISO=3). The attacker gains persistent, elevated access to the entire regulated infrastructure, justifying High likelihood and High impact.
T-M&O-CFG-COMP: Unauthorized or insecure configuration in Virtualisation orchestration/management systems														
V1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	**Low**	In UC-V1, M&O configuration errors (e.g. overly permissive API tokens or exposed management ports) affect a static, isolated environment managing non-critical VMs. The lack of multi-tenancy and external exposure limits the consequences of insecure defaults, keeping both likelihood and impact Low.
V2	2	2	3	2	2	2	2	2	2	2	11 → Med	10 → Med	**Medium**	In UC-V2, enterprise M&O systems are configured to manage multiple hypervisors with diverse role assignments, API scopes, and network policies. Insecure defaults (e.g. weak API tokens, unrestricted dashboards) or unauthorized configuration changes can propagate weaknesses across the entire enterprise cluster (CTL=3, ORB=2), giving Medium likelihood and Medium impact.
V3	2	3	3	2	3	3	3	3	3	3	13 → High	15 → High	**High**	In UC-V3, M&O configuration baselines enforce cluster-wide isolation policies, RBAC rules, and network segmentation for critical multi-tenant hypervisor fleets. A single insecure default (e.g. anonymous API access, overly broad role assignments) or unauthorized configuration change can cascade across all connected hypervisors and their workloads (ORB=3, CHG=3). This justifies High likelihood and High impact.
T-M&O-SC-COMP: Supply chain compromise in Virtualisation orchestration/management systems														
V1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-V1, a supply-chain compromise of an M&O component (e.g. API library, plugin module) affects only a single isolated management instance and its small set of non-critical VMs. The limited connectivity and minimal blast radius keep both likelihood and impact Low.

V2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-V2, enterprise M&O deployments rely on shared plugins, API libraries, and orchestration agents from various sources. A poisoned dependency embedded in the M&O control plane can introduce persistent backdoors affecting all hypervisors and VMs within the enterprise cluster. However, the impact is still bounded to one organizational tenant, giving Medium likelihood and Medium impact.
V3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	**High**	In UC-V3, M&O platforms depend on complex ecosystems of plugins, extensions, and API libraries. A supply-chain compromise of any control-plane component propagates to all connected hypervisors across multi-tenant, critical infrastructure (SUP=3, ORB=3). Because the M&O has administrative access to the entire fleet, the attacker gains the highest-privilege foothold possible, justifying High likelihood and High impact.

B.7.3 Risk Evaluation of CES use cases

B.7.3.1 CRS

Table B.7.3.1-1: CRS - Risk evaluation of CES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-CRS-INTF-ESC: Runtime-level container breakout														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-C1, the CRS runs on isolated devices or personal environments where containers host non-critical workloads and have limited external exposure. A runtime-level escape (e.g. via runc or namespace flaws) is technically possible, but any breakout only affects a single device and low-sensitivity applications, so both likelihood and impact are assessed as Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, the CRS underpins enterprise or edge container platforms that host multiple business microservices. A successful runtime-level escape allows a compromised container to interfere with peer containers on that node (and possibly across nodes, if combined with orchestration weaknesses), exposing all business workloads for that tenant. This justifies Medium likelihood and Medium impact.
C3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-C3, the CRS is part of high-assurance, multi-tenant container platforms processing critical or regulated workloads. Exploits that break container isolation (e.g. sandbox escapes, kernel namespace bugs) can lead to cross-tenant compromise, access to critical data, and potential safety or regulatory failures, so both likelihood and impact are High.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-CRS-INTEG-COMP: Runtime-level tampering														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	For UC-C1, local tampering with runtime binaries or configuration (e.g. disabling seccomp or cgroup enforcement) requires direct access to a single device and affects only non-critical applications. The orchestrator, if present, is minimal or absent, and the blast radius is small, so both likelihood and impact remain Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, the same CRS binaries and configs may be deployed across many enterprise nodes. If an attacker modifies runtime settings (e.g. disables AppArmor profiles, relaxes namespaces or cgroup limits), multiple business services lose isolation and resource controls. Although the effect is tenant-scoped, it impacts many workloads, leading to Medium likelihood and Medium impact.
C3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	High	In UC-C3, runtime integrity is a core assurance anchor for critical and regulated container workloads. Tampered runtime binaries or unverified state (e.g. modified seccomp profiles, spoofed namespaces) can silently bypass orchestrator policies across many tenants and nodes. Therefore, both likelihood and impact are rated High.
T-CRS-UNAUTH-ADMIN: Unauthorized runtime control access														
C1	1	1	2	1	1	1	1	1	1	1	6 → Low	5 → Low	Low	In UC-C1, the container runtime is typically driven by local commands (e.g. docker, ctr, podman) on a single, non-critical device. Misuse of these binaries via a local user or shared account is possible but affects only that device and low-value workloads, so likelihood and impact are Low.
C2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	In UC-C2, internal users or automation (CI/CD agents, deployment scripts) may have direct access to runtime control (e.g. privileged socket access or debug binaries). If authentication around such access is weak, insiders or compromised accounts can bypass orchestrator RBAC and start, stop, or modify containers directly on enterprise nodes. This yields Medium likelihood and Medium impact across the tenant's workloads.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	14 → High	High	In UC-C3, any exposed or weakly protected runtime control interface (e.g. an unprotected runtime socket, debug endpoint, or local root account on critical nodes) allows attackers to circumvent orchestrator policy and manipulate containers running critical/regulated workloads. In multi-tenant or federated scenarios, this can impact many tenants and critical services, hence High likelihood and High impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-CRS-UNAUTH-FUNC: Unauthorized privileged ops														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	Low	For UC-C1, users with legitimate but broad permissions (e.g. local "developer" accounts) can perform privileged runtime actions such as starting privileged containers or mounting host paths. However, these actions are confined to a single device and non-critical apps, so overall risk remains Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprise or edge environments often delegate limited runtime access to developers, operators, or automation tools. If authorization is insufficiently granular, these principals can escalate to privileged operations (e.g. modifying seccomp profiles, running containers as root, altering cgroup limits) and affect many containers belonging to that tenant, justifying Medium likelihood and Medium impact.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-C3, inadequate authorization in the CRS enables authenticated but mis-scoped identities to perform powerful runtime operations across nodes hosting critical or regulated workloads. This can override intended least-privilege policies, create new attack surfaces, or directly expose sensitive data, so both likelihood and impact are High.
T-CRS-DATA-DISC: Data disclosure via runtime														
C1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-C1, containers typically host low-sensitivity data (e.g. local telemetry, hobbyist applications). Even if CRS flaws allow access to container filesystems, mounts, or memory, the confidentiality impact is minimal, so both likelihood and impact are Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, the CRS may host business microservices processing confidential data (e.g. HR, analytics, internal APIs). Runtime-level weaknesses that expose mounted volumes, layered filesystems, or shared memory segments can leak sensitive internal data of the enterprise tenant, leading to Medium likelihood and Medium impact.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-C3, containers may handle financial, e-identity, or medical data in multi-tenant and federated setups. If runtime isolation fails, attackers can extract highly sensitive data from multiple tenants. This supports High likelihood and High impact.
T-CRS-AVAIL-DOS: Resource exhaustion														
C1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	Low	In UC-C1, a container that saturates CPU, memory, or I/O via weak cgroup settings will only affect a single device and non-critical services. The resulting DoS is limited to local functionality, so both likelihood and impact are Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprise CES nodes often run many business microservices. If the CRS does not enforce appropriate cgroup limits or priority settings, a single misbehaving or malicious container can starve other services on a node, causing noticeable but tenant-scoped outages, which is reflected as Medium likelihood and Medium impact.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
C3	3	3	2	2	3	3	3	3	3	2	13 → High	14 → High	High	In UC-C3, container platforms run critical or regulated services at scale. Insufficient isolation or resource governance in the CRS (e.g. no hard limits, shared resources across tenants) enables noisy-neighbour or deliberate resource exhaustion that can degrade or interrupt critical services across multiple tenants and edge sites, so both likelihood and impact are High.
T-CRS-LOG-INT: Logging weakness/tampering														
C1	1	1	1	1	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-C1, CRS logs (e.g. container start/stop events, seccomp denials) may be minimal or stored locally without protection. Attackers running inside containers could modify or delete such logs, but given the low sensitivity and small scale, the inability to reconstruct activity has limited impact, so risk remains Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprise CES platforms rely on CRS logs for troubleshooting and incident response (e.g. detecting abnormal restarts, denied syscalls, mount failures). Weak logging or easy log tampering delays detection of attacks against business microservices and complicates forensics, warranting Medium likelihood and Medium impact.
C3	2	3	3	2	3	2	3	3	3	3	13 → High	14 → High	High	In UC-C3, logging from the CRS is essential to maintain evidential integrity across critical workloads and tenants. If runtime logs are incomplete, not centrally collected, or modifiable by compromised containers, attackers can conduct stealthy operations (e.g. privilege escalation, forbidden syscalls) without trace across multi-tenant environments. This justifies High likelihood and High impact.
T-CRS-UPD-COMP: Malicious/compromised update														
C1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-C1, CRS updates (e.g. runc, containerd) are often applied manually on single devices, sometimes via direct downloads. A compromised package or replayed older runtime can weaken isolation on that device, but the effect is limited to one low-critical node, so likelihood is Low and impact Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprises may use repositories or CI/CD pipelines to distribute runtime updates. If these are compromised or unsigned, a malicious runtime can be rolled out across many nodes, weakening container isolation and seccomp/cgroup enforcement for all workloads of that tenant. This supports Medium likelihood and Medium impact.
C3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-C3, large-scale or automated update pipelines for CRS components mean that a poisoned runtime package can rapidly propagate across nodes hosting critical, multi-tenant workloads, effectively undermining isolation guarantees everywhere it is deployed. Hence both likelihood and impact are High.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-CRS-CFG-COMP: Insecure or Unauthorized configuration														
C1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	Low	In UC-C1, insecure runtime settings (e.g. no default seccomp/AppArmor profile, permissive host mounts) stem mainly from local experimentation or convenience. Misconfigurations are limited to one device and non-critical containers, so although isolation is reduced, overall risk remains Low.
C2	2	2	3	2	2	2	2	2	2	2	11 → Medium	10 → Medium	Medium	In UC-C2, runtime configuration (default seccomp profiles, namespace settings, resource limits) is often propagated via container engines or orchestrators. Weak defaults (e.g. unbounded CPU/memory, privileged containers, hostPath mounts) or unauthorized changes can significantly increase the attack surface across enterprise workloads, giving Medium likelihood and Medium impact.
C3	2	3	3	2	3	3	3	3	3	3	14 → High	15 → High	High	In UC-C3, CRS configuration is a primary control for maintaining strict isolation and resource separation in critical or regulated environments. Insecure or unauthorized runtime settings can directly enable container escape, lateral movement and high-impact DoS across tenants, so both likelihood and impact are High.
T-CRS-SC-COMP: Supply chain compromise														
C1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	Low	In UC-C1, a compromised runtime component (e.g. unpacker library, sandboxing module) affects a single isolated host. While it may permit container escape or tampering, the limited connectivity and low-value workloads keep the overall impact low; the Likelihood is Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Medium	10 → Medium	Medium	In UC-C2, enterprises may deploy CRS packages from shared repositories or base images. A poisoned dependency embedded in the runtime can introduce persistent backdoors or isolation weaknesses across many enterprise nodes, impacting all internal workloads but still within one organizational tenant, hence Medium likelihood and Medium impact.
C3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	High	In UC-C3, CRS supply-chain integrity is critical. Compromise of a low-level runtime dependency (e.g. syscall handling library, sandbox module) used in container platforms for multiple tenants means every container instantiation can execute malicious code with high privileges. This can enable systemic escape and cross-tenant compromise in public or federated CES, leading to High likelihood and High impact.

B.7.3.2 CE

Table B.7.3.2-1: CE – Risk evaluation of CES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-CE-INTF-ESC: Engine-level cross-container interference														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-C1, the container engine runs on an isolated device or personal environment where containers host non-critical workloads and have limited external exposure. Engine-level interference between containers (e.g. via API misuse or plugin vulnerabilities) is technically possible, but any cross-container impact only affects a single device and low-sensitivity applications, so both likelihood and impact are assessed as Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, the container engine underpins enterprise or edge platforms hosting multiple business microservices. A successful engine-level interference allows a compromised container to affect peer containers sharing the same engine (and potentially other nodes, if combined with orchestration weaknesses), exposing all business workloads for that tenant. This justifies Medium likelihood and Medium impact.
C3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	**High**	In UC-C3, the container engine is part of a high-assurance, multi-tenant container platform processing critical or regulated workloads. Engine-level interference exploiting isolation weaknesses in the engine's API, plugin stack, or configuration can lead to cross-container compromise, access to critical data, and potential safety or regulatory failures, so both likelihood and impact are High.
T-CE-INTEG-COMP: Integrity compromise of container engine and managed images														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	For UC-C1, tampering with the container engine binaries, configuration files, or locally cached container images requires direct access to a single device and affects only non-critical applications. The orchestrator, if present, is minimal or absent, and the blast radius is small, so both likelihood and impact remain Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, the same container engine binaries and configurations may be deployed across many enterprise nodes. If an attacker modifies engine binaries or cached image layers (e.g. disabling security controls, injecting malicious startup logic), multiple business services lose integrity guarantees. Although the effect is tenant-scoped, it impacts many workloads, leading to Medium likelihood and Medium impact.
C3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	**High**	In UC-C3, container engine integrity is a core assurance anchor for critical and regulated containerized workloads. Tampered engine binaries or manipulated image layers can silently execute malicious logic across many tenants and nodes without being detected by the orchestrator. Therefore, both likelihood and impact are rated High.

T-CE-UNAUTH-ADMIN: Unauthorized access to container engine control endpoints														
C1	1	1	2	1	1	1	1	1	1	1	6 → Low	5 → Low	**Low**	In UC-C1, the container engine is typically driven by local commands (e.g. docker, nerdctl, podman) on a single, non-critical device. The control socket or API is not exposed remotely, and access is confined to local users. Even if socket permissions are overly broad, the impact is limited to a single device and low-value workloads, so likelihood and impact are Low.
C2	2	2	3	2	2	2	2	2	2	2	11 → Med	10 → Med	**Medium**	In UC-C2, internal users or automation (CI/CD agents, deployment scripts) may have access to the container engine's control socket or remote API endpoint. If authentication around such access is weak (CTL=3), insiders or compromised accounts can bypass orchestrator RBAC and start, stop, or modify containers directly on enterprise nodes. This yields Medium likelihood and Medium impact across the tenant's workloads.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	**High**	In UC-C3, any exposed or weakly protected engine control interface (e.g. an unauthenticated daemon socket, exposed gRPC endpoint, or local root account on critical nodes) allows attackers to circumvent orchestrator policy and directly manipulate containers running critical or regulated workloads. In multi-tenant scenarios, this can impact many tenants and critical services, hence High likelihood and High impact.
T-CE-UNAUTH-FUNC: Unauthorized privileged operations or data access in the container engine														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	For UC-C1, users with legitimate but broad permissions (e.g. local developer accounts with access to the container socket) can perform privileged engine operations such as starting privileged containers or mounting host paths. However, these actions are confined to a single device and non-critical applications, so overall risk remains Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, enterprise or edge environments often delegate limited engine access to developers, operators, or automation tools. If authorization is insufficiently granular, these principals can escalate to privileged engine operations (e.g. modifying daemon configurations, running containers as root, altering network driver settings) and affect many containers belonging to that tenant, justifying Medium likelihood and Medium impact.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	**High**	In UC-C3, inadequate authorization in the container engine enables authenticated but mis-scoped identities to perform powerful operations across nodes hosting critical or regulated workloads. This can override intended least-privilege policies, create new attack surfaces, or directly expose sensitive data, so both likelihood and impact are High.
T-CE-DATA-DISC: Unauthorized disclosure of sensitive data via the container engine														
C1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	**Low**	In UC-C1, containers typically host low-sensitivity data (e.g. local telemetry, hobbyist applications). Even if engine weaknesses allow access to local image layers, container filesystems, or host-mounted paths, the confidentiality impact is minimal, so both likelihood and impact are Low.

C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, the container engine may host business microservices processing confidential data (e.g. HR, analytics, internal APIs). Engine-level weaknesses that expose cached image layers, container secrets, or locally stored volumes can leak sensitive internal data of the enterprise tenant, leading to Medium likelihood and Medium impact.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	**High**	In UC-C3, containers may handle financial, e-identity, or medical data in multi-tenant and federated setups. If engine-level isolation fails or secrets stored in image layers or engine configurations are exposed, attackers can extract highly sensitive data from multiple tenants. This supports High likelihood and High impact.
T-CE-AVAIL-DOS: DoS via resource exhaustion in the container engine														
C1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	**Low**	In UC-C1, an attacker exploiting the engine's local API or socket to trigger excessive image pulls, container starts/stops, or logging operations will only affect a single device and non-critical services. The resulting DoS is limited to local functionality, so both likelihood and impact are Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, enterprise CES nodes often run many business microservices. If the container engine does not enforce rate limits or resource quotas on API operations, a single misbehaving or malicious container (or a compromised CI/CD agent) can saturate the engine's processing capacity, causing container lifecycle delays and noticeable tenant-scoped outages. This is reflected as Medium likelihood and Medium impact.
C3	3	3	2	2	3	3	3	3	3	2	13 → High	14 → High	**High**	In UC-C3, container engines serve critical or regulated services at scale. Insufficient API rate limiting or resource controls in the engine (e.g. unconstrained image pull storms, excessive daemon restarts) enables noisy-neighbour or deliberate resource exhaustion that can degrade or interrupt critical services across multiple tenants and edge sites, so both likelihood and impact are High.
T-CE-LOG-INT: Insufficient logging or log tampering in the Container Engine														
C1	1	1	1	1	1	1	1	1	1	2	5 → Low	6 → Low	**Low**	In UC-C1, container engine logs (e.g. image pull events, container start/stop, API calls) may be minimal or stored locally without protection. Attackers with local access could modify or delete such logs, but given the low sensitivity and small scale, the inability to reconstruct activity has limited impact, so risk remains Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, enterprise CES platforms rely on engine logs for troubleshooting and incident response (e.g. detecting abnormal image pulls, unauthorized container modifications, or daemon errors). Weak logging or easy log tampering at the engine level delays detection of attacks against business microservices and complicates forensics, warranting Medium likelihood and Medium impact.

C3	2	3	3	2	3	2	3	3	3	3	13 → High	14 → High	**High**	In UC-C3, logging from the container engine is essential to maintain evidential integrity across critical workloads and tenants. If engine logs are incomplete, not centrally collected, or modifiable by compromised containers or operators, attackers can conduct stealthy operations (e.g. unauthorized image replacements, privilege escalation, forbidden capabilities) without trace across multi-tenant environments. This justifies High likelihood and High impact.
T-CE-UPD-COMP: Malicious or compromised update in the Container Engine														
C1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-C1, container engine updates (e.g. Docker Engine, containerd, Podman) are often applied manually on single devices, sometimes via direct downloads. A compromised package or replayed older engine binary can weaken isolation on that device, but the effect is limited to one low-critical node, so likelihood and impact are Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, enterprises may use repositories or CI/CD pipelines to distribute container engine updates. If these are compromised or unsigned, a malicious engine binary can be rolled out across many nodes, weakening container isolation, disabling security controls, and affecting all business workloads of that tenant. This supports Medium likelihood and Medium impact.
C3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	**High**	In UC-C3, large-scale or automated update pipelines for the container engine mean that a poisoned engine package can rapidly propagate across nodes hosting critical, multi-tenant workloads, effectively undermining isolation guarantees everywhere it is deployed. Because the engine directly interfaces with the host OS and launches containers, a compromised update can execute arbitrary code with elevated privileges, hence both likelihood and impact are High.
T-CE-CFG-COMP: Unauthorized or insecure configuration in the Container Engine														
C1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	**Low**	In UC-C1, insecure engine defaults (e.g. an exposed API socket without TLS/authentication, unrestricted default bridge networking, or permission to run privileged containers) affect only a single device and non-critical containers. The lack of multi-tenancy and external exposure keeps likelihood and impact Low.
C2	2	2	3	2	2	2	2	2	2	2	11 → Med	10 → Med	**Medium**	In UC-C2, enterprise container engine configurations (Docker daemon JSON, containerd config, runtime parameters) are often propagated via orchestrators or automated tooling. Weak defaults or unauthorized configuration changes (e.g. enabling host networking, disabling user namespace remapping, exposing the daemon socket) can significantly increase the attack surface across enterprise workloads, giving Medium likelihood and Medium impact.

C3	2	3	3	2	3	3	3	3	3	3	13 → High	15 → High	**High**	In UC-C3, container engine configuration is a primary control for maintaining strict isolation and resource separation in critical or regulated environments. Insecure or unauthorized engine settings can directly enable container escape, lateral movement, and high-impact DoS across tenants. Because the engine mediates between containers, the container runtime stack, and the host OS, insecure configurations at this layer can undermine both workload security and host-level protections, so both likelihood and impact are High.
T-CE-SC-COMP: Supply chain compromise in a Container Engine														
C1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-C1, a compromised engine dependency (e.g. registry client library, image build tool, or network driver) affects a single isolated host. While it may permit container escape or image manipulation, the limited connectivity and low-value workloads keep the overall impact low; the likelihood is also Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, enterprises may deploy container engine packages from shared repositories or base images. A poisoned dependency embedded in the engine can introduce persistent backdoors or isolation weaknesses across many enterprise nodes, impacting all internal workloads but still within one organizational tenant, hence Medium likelihood and Medium impact.
C3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	**High**	In UC-C3, container engine supply-chain integrity is critical. Compromise of a dependency (e.g. registry client library, image unpacker, or network driver) used in container platforms for multiple tenants means every container operation can execute malicious code at elevated privileges, potentially enabling systemic escape and cross-tenant compromise in public or federated CES, leading to High likelihood and High impact.

B.7.3.3 CO

Table B.7.3.3-1: CO - Risk evaluation of CES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
T-CO-INTF-ESC: Orchestrator-level cross-workload interference														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-C1, the container orchestrator manages a single isolated device or personal environment with few non-critical containers and limited administrative exposure (EXT=1, TEN=1). Orchestration misconfigurations or policy gaps that allow cross-namespace interference are technically possible, but any resulting impact only affects low-value workloads on a single node, so both likelihood and impact are assessed as Low.

C2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, the container orchestrator governs enterprise or edge container platforms hosting multiple business microservices across namespaces or projects. An attacker with access to orchestration APIs can exploit policy gaps or network segmentation flaws to interfere with workloads managed by the orchestrator (ISO=2, ORB=2), exposing all business workloads for that tenant. This justifies Medium likelihood and Medium impact.
C3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	**High**	In UC-C3, the orchestrator controls a high-assurance, multi-tenant container platform processing critical or regulated workloads. Exploitation of orchestration misconfigurations, RBAC gaps, or network policy weaknesses can lead to cross-namespace or cross-tenant interference, access to critical data, and potential safety or regulatory failures, so both likelihood and impact are High.
T-CO-INTEG-TRUST: Orchestrator trusts compromised nodes or components														
C1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	**Low**	In UC-C1, the orchestrator manages a single static host with limited node attestation requirements. If the container host is compromised and feeds falsified state data, the impact remains limited to a single low-value environment. The minimal orchestration blast radius (ORB=1) and low data sensitivity (DAT=1) keep both likelihood and impact Low.
C2	1	2	3	2	2	2	2	2	2	3	10 → Med	11 → Med	**Medium**	In UC-C2, an enterprise orchestrator may schedule workloads across multiple internal container hosts or engine instances without rigorous per-node attestation. A compromised node that sends falsified health or capacity data can influence scheduling decisions, potentially exposing enterprise workloads to a tampered execution environment (ORB=2, ISO=2). Some enterprise monitoring may partially compensate (DRC=3). Both likelihood and impact are Medium.
C3	3	3	3	3	3	3	2	3	3	3	15 → High	14 → High	**High**	In UC-C3, high-assurance container orchestration relies on trusted attestation of each container host and engine in the fleet. If attestation mechanisms are absent or can be spoofed by a compromised node, the orchestrator may continue scheduling regulated or critical multi-tenant workloads onto a tampered host (TEN=3, DAT=3, ORB=3). The attacker gains a privileged position over all workloads placed on that node without triggering orchestrator-level alerts, giving High likelihood and High impact.
T-CO-UNAUTH-ADMIN: Unauthorized access to orchestrator management plane														
C1	1	1	2	2	2	1	1	1	1	2	8 → Low	6 → Low	**Low**	In UC-C1, the orchestrator's management interfaces are reachable only via local or isolated networks and control a small number of non-critical containers on a single node (EXT=1, TEN=1). Even if authentication is weak, the lack of external exposure and the minimal value of managed workloads keep both likelihood and impact Low.

C2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, the orchestrator's management APIs and dashboards are reachable from the enterprise network and govern all container nodes and workloads in the cluster (CTL=3, ORB=2). Unauthorized access via compromised credentials, missing MFA, or default accounts gives an attacker full cluster control over internal workloads. The effect is scoped to one organizational tenant but can impact many microservices, giving Medium likelihood and Medium impact.
C3	3	3	3	3	3	3	3	3	3	3	15 → High	15 → High	**High**	In UC-C3, the orchestrator's administrative interfaces provide cluster-wide control over critical multi-tenant container fleets (TEN=3, ORB=3). External or broadly reachable control planes (EXT=3) combined with potentially weak authentication or compromised admin credentials represent the highest-value attack target. A successful breach gives the attacker unrestricted cluster control over all hosted workloads and regulated data, making both likelihood and impact High.
T-CO-UNAUTH-FUNC: Unauthorized privileged operations or data access in the orchestrator														
C1	1	1	2	1	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-C1, the orchestrator's users or automation tools have limited roles managing a small set of non-critical containers. Even if RBAC is misconfigured, the privilege abuse can only affect that single node's workloads, and the environment is not exposed to untrusted users, so both likelihood and impact are Low.
C2	1	2	3	1	3	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, enterprise orchestrator deployments often delegate limited API access to developers, CI/CD pipelines, and automation tools. Insufficient RBAC granularity (CTL=3, PRV=3) can allow a compromised service account or low-privilege token to perform cluster-level operations (e.g. creating privileged pods, modifying admission policies, accessing secrets) affecting all enterprise workloads in the cluster (ORB=2), giving Medium likelihood and Medium impact.
C3	2	3	2	3	3	3	3	3	3	3	13 → High	15 → High	**High**	In UC-C3, the orchestrator grants broad privileged access to automation systems, multi-team operators, and federated administrators managing critical or regulated workloads. Weak or inconsistent RBAC enforcement allows over-privileged identities to execute high-impact operations across many tenants (TEN=3, ORB=3), such as modifying network policies, deploying unverified workloads, or accessing restricted secrets. This drives both likelihood and impact to High.
T-CO-DATA-DISC: Unauthorized disclosure of sensitive data via the container orchestrator														
C1	1	1	1	1	1	1	1	1	1	2	5 → Low	6 → Low	**Low**	In UC-C1, the orchestrator manages a small number of non-critical containers on an isolated host. Orchestration state, container configurations, or stored secrets have low sensitivity (DAT=1), and the management plane is not externally exposed (EXT=1). Even if access controls are weak, the resulting confidentiality impact is minimal, so both likelihood and impact are Low.

C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, the orchestrator aggregates sensitive information about enterprise workloads - including container images, secrets, network topology, and API credentials. Vulnerabilities or access-control weaknesses in the management API, dashboard, or CLI can expose this data to internal adversaries, affecting the entire enterprise tenant. The scope is tenant-limited but significantly impacts business operations, giving Medium likelihood and impact.
C3	3	3	3	2	3	3	3	3	3	3	14 → High	15 → High	**High**	In UC-C3, the orchestrator handles multi-tenant workload configurations, secrets, service credentials, and potentially runtime telemetry for regulated or safety-critical services (DAT=3). A breach of orchestrator data stores or API outputs gives attackers a comprehensive map of the entire infrastructure alongside direct access to sensitive secrets across tenants (ISO=3, ORB=3). This supports High likelihood and High impact.
T-CO-AVAIL-DOS: DoS via resource exhaustion in the container orchestrator														
C1	1	1	1	1	1	1	1	1	1	1	5 → Low	5 → Low	**Low**	In UC-C1, the orchestrator manages a single host with a small number of non-critical containers. Flooding the orchestration API or control loop with excessive requests would at most delay scheduling or management operations on that single node. Given the limited scope and non-critical workloads, both likelihood and impact are Low.
C2	1	2	2	2	2	2	2	2	2	2	9 → Med	10 → Med	**Medium**	In UC-C2, the orchestrator's schedulers and APIs handle ongoing lifecycle operations for multiple enterprise containers. Excessive scheduling requests, frequent scaling operations, or malformed deployment manifests can render management operations unresponsive and prevent timely maintenance, patching, or incident response across the enterprise tenant. Both likelihood and impact are Medium.
C3	2	3	3	2	3	3	3	2	3	2	13 → High	13 → High	**High**	In UC-C3, orchestration control planes govern critical multi-tenant container fleets handling regulated workloads (TEN=3, ORB=3). An attacker who floods orchestration APIs, generates excessive pod lifecycle events, or exploits autoscaling policies can paralyse the scheduling layer, preventing timely workload migration, failover, or maintenance across the entire cluster. This can directly degrade service availability for critical services, resulting in High likelihood and High impact.
T-CO-LOG-INT: Insufficient logging or log tampering in the Container Orchestrator														
C1	1	1	1	1	1	1	1	1	1	2	5 → Low	6 → Low	**Low**	In UC-C1, orchestrator event and audit logs may be minimal or stored locally without protection. Even if logs are incomplete or tampered, the low data sensitivity and minimal orchestration scope (DAT=1, ORB=1) limit the forensic impact, so overall risk remains Low.

C2	1	2	2	2	2	2	1	2	2	2	9 → Med	9 → Med	**Medium**	In UC-C2, the orchestrator is the primary source of lifecycle and administrative audit records for enterprise container clusters. Insufficient audit trails or easy log modification can delay detection of targeted configuration changes, unauthorized deployments, or RBAC policy modifications within the enterprise environment. This warrants Medium likelihood and Medium impact.
C3	2	3	3	2	3	2	2	3	3	3	13 → High	13 → High	**High**	In UC-C3, the orchestrator is the central source of audit truth for critical multi-tenant container infrastructure. Tampering with or suppressing orchestrator logs eliminates evidence of cross-tenant access violations, unauthorized workload deployments, or privilege escalations affecting regulated services (DAT=3, ORB=3). The inability to reconstruct attack timelines also hampers breach containment and regulatory reporting, resulting in High likelihood and High impact.
T-CO-UPD-COMP: Malicious or compromised update in the Container Orchestrator														
C1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-C1, orchestrator software updates are typically performed manually or infrequently and affect only a single node managing non-critical containers. A malicious update could compromise the orchestration function on that node, but the limited scale and low workload sensitivity keep both likelihood and impact Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, enterprise orchestrator platforms may be updated via automated pipelines. A compromised update to the control plane components (API server, scheduler, controller manager) or their plugins can be rolled out across the enterprise cluster (SUP=2, ORB=2), potentially allowing the attacker to modify workload scheduling, bypass RBAC rules, or deploy malicious workloads. This supports Medium likelihood and impact.
C3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	**High**	In UC-C3, the orchestrator is managed at scale with highly automated update pipelines (SUP=3). A compromised update to the orchestration control plane propagates immediately to all connected container nodes and workloads across critical, multi-tenant deployments (ORB=3, ISO=3). The attacker gains persistent, cluster-wide access to the entire regulated infrastructure, justifying High likelihood and High impact.
T-CO-CFG-COMP: Unauthorized or insecure configuration in the Container Orchestrator														
C1	1	1	2	1	2	1	1	1	1	1	7 → Low	5 → Low	**Low**	In UC-C1, orchestrator configuration errors (e.g. anonymous API access, default administrator accounts, or permissive admission controller policies) affect a static, isolated environment managing non-critical containers. The lack of multi-tenancy and external exposure limits the consequences of insecure defaults, keeping both likelihood and impact Low.

C2	1	2	3	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, enterprise orchestrators are configured with diverse role assignments, API scopes, admission policies, and network segmentation rules. Insecure defaults (e.g. unrestricted inter-pod communication, overly permissive RBAC roles) or unauthorized configuration changes can propagate weaknesses across the entire enterprise cluster (CTL=3, ORB=2), giving Medium likelihood and Medium impact.
C3	2	3	3	2	3	3	3	3	3	3	13 → High	15 → High	**High**	In UC-C3, orchestrator configuration baselines enforce cluster-wide isolation policies, network segmentation, admission controls, and RBAC rules for critical multi-tenant container fleets. A single insecure default (e.g. anonymous API access, disabled pod security admission, unrestricted network policies) or unauthorized configuration change can cascade across all connected container nodes and their workloads (ORB=3, CHG=3). This justifies High likelihood and High impact.
T-CO-SC-COMP: Supply chain compromise in a Container Orchestrator														
C1	1	1	1	2	2	1	1	1	1	2	7 → Low	6 → Low	**Low**	In UC-C1, a supply-chain compromise of an orchestrator component (e.g. API library, scheduler plugin, or module) affects only a single isolated management instance and its small set of non-critical containers. The limited connectivity and minimal blast radius keep both likelihood and impact Low.
C2	2	2	2	2	2	2	2	2	2	2	10 → Med	10 → Med	**Medium**	In UC-C2, enterprise orchestrator deployments rely on shared plugins, API libraries, and orchestration agents from various sources. A poisoned dependency embedded in the orchestrator control plane can introduce persistent backdoors affecting all container nodes and workloads within the enterprise cluster. However, the impact is still bounded to one organizational tenant, giving Medium likelihood and Medium impact.
C3	2	3	2	3	3	2	3	3	3	3	13 → High	14 → High	**High**	In UC-C3, orchestrator platforms depend on complex ecosystems of plugins, extensions, and API libraries. A supply-chain compromise of any control-plane component propagates to all connected container nodes across multi-tenant, critical infrastructure (SUP=3, ORB=3). Because the orchestrator controls all cluster workloads and nodes, the compromise propagates to the entire environment, justifying High likelihood and High impact.

B.7.4 Risk Evaluation of common hardware-level threat T-ALL-HW-ACCESS

The threat T-ALL-HW-ACCESS (Compromise via hardware-level or platform trust components) is common to both Virtualisation Execution Stacks (VES) and Container Execution Stacks (CES).

It models compromise of the underlying hardware or platform trust components in a way that can bypass or undermine software-based security mechanisms, including boot integrity verification, runtime integrity checks, attestation, and isolation controls.

Because hardware-level compromise typically requires a high level of attacker sophistication and is heavily dependent on platform supply-chain assurance, the likelihood remains relatively low to medium, while the impact can be catastrophic for high-assurance, multi-tenant or critical deployments.

Table B.7.4-1 applies the risk assessment methodology of Annex B to this threat for the VES and CES use cases.

Table B.7.4-1: Risk evaluation of the common hardware-level threat T-ALL-HW-ACCESS across VES and CES use cases

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
V1	1	1	1	2	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-V1, hypervisors run on isolated hosts with a small number of non-critical VMs and limited connectivity. Hardware-level compromise (e.g. tampered firmware or physical access) would give full control of that single host, but the data is low-sensitivity and there is no large orchestration blast radius (ISO=1, DAT=1, ORB=1). Supply-chain risk exists (SUP=2), but exposure and attacker incentive (EXT=1, TEN=1, CTL=1, PRV=1) are low. Overall likelihood and impact are therefore assessed as Low.
V2	2	2	2	2	1	2	2	2	2	2	9 → Medium	10 → Medium	Medium	In UC-V2, Type-1/Type-2 hypervisors support general business workloads in enterprise/private clouds and connected edge clusters. Commodity hardware and shared infrastructure increase the relevance of hardware supply-chain issues (EXT=2, TEN=2, SUP=2), but the attack still requires high sophistication (PRV=1). A successful hardware-level compromise affects multiple business VMs and nodes within one organization (ISO=2, DAT=2, ORB=2), and standard enterprise monitoring/recovery exists (DRC=2), so both likelihood and impact are Medium.
V3	3	3	2	3	1	3	3	3	3	3	12 → Medium	15 → High	High	In UC-V3, hypervisors are the foundation of high-assurance, multi-tenant infrastructures processing critical or regulated workloads. These platforms are high-value targets for sophisticated hardware-level attacks (EXT=3, TEN=3, SUP=3) even though such attacks remain complex (PRV=1). A compromised CPU, firmware, or hardware security module can bypass secure boot, attestation, and isolation controls, impacting all tenants and critical services across clusters (ISO=3, DAT=3, ORB=3). Detection and recovery from deep hardware compromise are limited (DRC=3). As a result, likelihood is Medium but impact is High, and the overall risk level is High.
C1	1	1	1	2	1	1	1	1	1	2	6 → Low	6 → Low	Low	In UC-C1, the CES run on devices with limited connectivity (e.g. hobbyist setups, isolated test/dev hosts) that execute non-critical containers. Hardware-level compromise would allow full control of the host and all containers, but the workloads and data are low-sensitivity and there is no meaningful multi-tenant blast radius (ISO=1, DAT=1, ORB=1). Supply-chain risk exists (SUP=2), but exposure and tenant multiplicity are low (EXT=1, TEN=1, CTL=1, PRV=1), so both likelihood and impact are Low.

UC	EXT	TEN	CTL	SUP	PRV	CHG	ISO	DAT	ORB	DRC	Likelihood (sum/level)	Impact (sum/level)	Risk	Rationale
C2	2	2	2	2	1	2	2	2	2	2	9 → Medium	10 → Medium	Medium	In UC-C2, the CES supports general business microservices and edge workloads. A hardware-level compromise of the underlying platform (e.g. compromised firmware or SoC) affects all CRS instances and containers on affected nodes, exposing sensitive internal data and services within one organization (ISO=2, DAT=2, ORB=2). Commodity hardware and common supply chains (SUP=2) increase the exposure compared to UC-C1, but attacks still require high sophistication (PRV=1). This results in Medium likelihood and Medium impact.
C3	3	3	2	3	1	3	3	3	3	3	12 → Medium	15 → High	High	In UC-C3, the CES underpins critical or regulated containerized workloads in public, federated, or zero-trust environments, including edge sites performing safety-relevant processing. Hardware-level compromise of CPUs, firmware or platform trust components can silently bypass all CRS and orchestrator controls, enabling arbitrary access to containers and data across many tenants and sites (ISO=3, DAT=3, ORB=3). These platforms are attractive targets for well-resourced attackers (EXT=3, TEN=3, SUP=3), and deep hardware compromise is hard to detect and remediate (DRC=3). Accordingly, likelihood is Medium but impact is High, giving an overall High risk.

B.8 Threat-to-Requirement Traceability Matrix

B.8.1 VES

B.8.1.1 Hypervisor

Table B.8.1.1-1: Hypervisor - threats to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	**Low**	**Medium**	**High**
T-HYP1-INTF-ESC/ T-HYP2-HOST-ABUSE	REQ-H-VM-ISO-001 REQ-H-CP-ISO-001 REQ-H-NP-ISO-001	REQ-H-VM-ISO-002 REQ-H-CP-ISO-002 REQ-H-NP-ISO-002	REQ-H-VM-ISO-003 REQ-H-VM-ISO-004 REQ-H-CP-ISO-003 REQ-H-NP-ISO-003
T-HYP1-INTEG-COMP/T-HYP2-INTEG-COMP	REQ-H-IMG-INT-001	REQ-H-B-INT-002 REQ-H-IMG-INT-002 REQ-H-RP-INT-002	REQ-H-B-INT-002 REQ-H-IMG-INT-003 REQ-H-RP-INT-002 REQ-H-RA-INT-003 REQ-H-RA-INT-004
T-HYP1-UNAUTH-ADMIN/ T-HYP2-UNAUTH-ADMIN	REQ-H-ADMIN-AUTH-001 REQ-H-SERV-AUTH-001	REQ-H-ADMIN-AUTH-002 REQ-H-SERV-AUTH-002	REQ-H-ADMIN-AUTH-003 REQ-H-SERV-AUTH-003
T-HYP1-UNAUTH-FUNC/ T-HYP2-UNAUTH-FUNC	REQ-H-ADMIN-AUTHZ-001 REQ-H-SERV-AUTHZ-001	REQ-H-ADMIN-AUTHZ-002 REQ-H-SERV-AUTHZ-002	REQ-H-ADMIN-AUTHZ-003 REQ-H-SERV-AUTHZ-003
T-HYP1-DATA-DISC/ T-HYP2-DATA-DISC	REQ-H-CONF-001 REQ-H-DM-001 REQ-H-DM-002 REQ-H-DECOM-001	REQ-H-CONF-002 REQ-H-CONF-003	REQ-H-CONF-004
T-HYP1-AVAIL-DOS/ T-HYP2-AVAIL-DOS	REQ-H-AVAIL-001	REQ-H-AVAIL-002	REQ-H-AVAIL-003
T-HYP1-LOG-INT/ T-HYP2-LOG-INT	REQ-H-LOG-001	REQ-H-LOG-002	REQ-H-LOG-003
T-HYP1-UPD-COMP/ T-HYP2-UPD-COMP	REQ-H-UPD-001	REQ-H-UPD-002	REQ-H-UPD-003
T-HYP1-CFG-COMP/ T-HYP2-CFG-COMP	REQ-H-CFG-001	REQ-H-CFG-002	REQ-H-CFG-003
T-HYP1-SC-COMP/ T-HYP2-SC-COMP	REQ-H-IMG-INT-001 REQ-H-UPD-001	REQ-H-B-INT-002 REQ-H-IMG-INT-002 REQ-H-RP-INT-002 REQ-H-UPD-002	REQ-H-B-INT-002 REQ-H-IMG-INT-003 REQ-H-RP-INT-002 REQ-H-RA-INT-003 REQ-H-RA-INT-004 REQ-H-UPD-003

B.8.1.2 M&O System

Table B.8.1.2-1: M&O system - threats to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	**Low**	**Medium**	**High**
T-M&O-CTRL-PLANE	REQ-M&O-AUTHZ-001 REQ-M&O-CFG-001 REQ-M&O-COM-001 REQ-M&O-LOG-001	REQ-M&O-AUTHZ-002 REQ-M&O-CFG-002 REQ-M&O-COM-002 REQ-M&O-LOG-002	REQ-M&O-AUTHZ-003 REQ-M&O-CFG-003 REQ-M&O-COM-003 REQ-M&O-LOG-003
T-M&O-INTEG-TRUST	REQ-M&O-COM-001 REQ-M&O-INT-001	REQ-M&O-COM-002 REQ-M&O-INT-002	REQ-M&O-COM-003 REQ-M&O-CFG-003 REQ-M&O-INT-003
T-M&O-UNAUTH-ADMIN	REQ-M&O-AUTH-001 REQ-M&O-LOG-001	REQ-M&O-AUTH-002 REQ-M&O-LOG-002	REQ-M&O-AUTH-003 REQ-M&O-LOG-003
T-M&O-UNAUTH-FUNC	REQ-M&O-AUTHZ-001 REQ-M&O-CFG-001 REQ-M&O-LOG-001	REQ-M&O-AUTHZ-002 REQ-M&O-CFG-002 REQ-M&O-LOG-002	REQ-M&O-AUTHZ-003 REQ-M&O-CFG-003 REQ-M&O-LOG-003
T-M&O-DATA-DISC	REQ-M&O-AUTHZ-001 REQ-M&O-COM-001 REQ-M&O-DECOM-001 REQ-M&O-DECOM-002	REQ-M&O-AUTHZ-002 REQ-M&O-COM-002 REQ-M&O-CFG-002	REQ-M&O-AUTHZ-003 REQ-M&O-COM-003 REQ-M&O-CFG-003
T-M&O-AVAIL-DOS	REQ-M&O-CFG-001 REQ-M&O-LOG-001	REQ-M&O-CFG-002 REQ-M&O-LOG-002	REQ-M&O-CFG-003 REQ-M&O-LOG-003
T-M&O-LOG-INT	REQ-M&O-LOG-001	REQ-M&O-LOG-002	REQ-M&O-LOG-003
T-M&O-UPD-COMP	REQ-M&O-INT-001 REQ-M&O-UPD-001 REQ-M&O-LOG-001	REQ-M&O-INT-002 REQ-M&O-UPD-002 REQ-M&O-LOG-002	REQ-M&O-INT-003 REQ-M&O-UPD-003 REQ-M&O-LOG-003
T-M&O-CFG-COMP	REQ-M&O-CFG-001 REQ-M&O-AUTHZ-001 REQ-M&O-LOG-001	REQ-M&O-CFG-002 REQ-M&O-AUTHZ-002 REQ-M&O-LOG-002	REQ-M&O-CFG-003 REQ-M&O-AUTHZ-003 REQ-M&O-LOG-003
T-M&O-SC-COMP	REQ-M&O-INT-001 REQ-M&O-UPD-001	REQ-M&O-INT-002 REQ-M&O-UPD-002	REQ-M&O-INT-003 REQ-M&O-UPD-003

B.8.2 CES

B.8.2.1 CRS

Table B.8.2.1-1: CRS - threats to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	**Low**	**Medium**	**High**
T-CRS-INTF-ESC	REQ-CRS-CN-ISO-001 REQ-CRS-CP-ISO-001 REQ-CRS-NP-ISO-001	REQ-CRS-CN-ISO-002 REQ-CRS-CP-ISO-002 REQ-CRS-NP-ISO-002	REQ-CRS-CN-ISO-003 REQ-CRS-CP-ISO-003 REQ-CRS-NP-ISO-003
T-CRS-INTEG-COMP	REQ-CRS-IMG-INT-001	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-002 REQ-CRS-RP-INT-002	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-003 REQ-CRS-RP-INT-002 REQ-CRS-RA-INT-003 REQ-CRS-RA-INT-004
T-CRS-UNAUTH-ADMIN	REQ-CRS-ADMIN-AUTH-001 REQ-CRS-SERV-AUTH-001	REQ-CRS-ADMIN-AUTH-002 REQ-CRS-SERV-AUTH-002	REQ-CRS-ADMIN-AUTH-003 REQ-CRS-SERV-AUTH-003
T-CRS-UNAUTH-FUNC	REQ-CRS-ADMIN-AUTHZ-001 REQ-CRS-SERV-AUTHZ-001	REQ-CRS-ADMIN-AUTHZ-002 REQ-CRS-SERV-AUTHZ-002	REQ-CRS-ADMIN-AUTHZ-003 REQ-CRS-SERV-AUTHZ-003
T-CRS-DATA-DISC	REQ-CRS-CONF-001 REQ-CRS-DM-001 REQ-CRS-DM-002 REQ-CRS-DECOM-001	REQ-CRS-CONF-002 REQ-CRS-CONF-003	REQ-CRS-CONF-004
T-CRS-AVAIL-DOS	REQ-CRS-AVAIL-001	REQ-CRS-AVAIL-002 REQ-CRS-AVAIL-003	REQ-CRS-AVAIL-004
T-CRS-LOG-INT	REQ-CRS-LOG-001	REQ-CRS-LOG-002	REQ-CRS-LOG-003
T-CRS-UPD-COMP	REQ-CRS-UPD-001	REQ-CRS-UPD-002	REQ-CRS-UPD-003
T-CRS-CFG-COMP	REQ-CRS-CFG-001	REQ-CRS-CFG-002	REQ-CRS-CFG-003
T-CRS-SC-COMP	REQ-CRS-IMG-INT-001 REQ-CRS-UPD-001	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-002 REQ-CRS-RP-INT-002 REQ-CRS-UPD-002	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-003 REQ-CRS-RP-INT-002 REQ-CRS-RA-INT-003 REQ-CRS-RA-INT-004 REQ-CRS-UPD-003

B.8.2.2 CE

Table B.8.2.2-1: CE - threats to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	**Low**	**Medium**	**High**
T-CE-INTF-ESC	REQ-CE-ISO-001	REQ-CE-ISO-002	REQ-CE-ISO-003
T-CE-INTEG-COMP	REQ-CE-INTEG-001 REQ-CE-UPD-001	REQ-CE-INTEG-002 REQ-CE-UPD-002	REQ-CE-INTEG-003
T-CE-UNAUTH-ADMIN	REQ-CE-AUTH-001	REQ-CE-AUTH-002	REQ-CE-AUTH-003
T-CE-UNAUTH-FUNC	REQ-CE-AUTHZ-001	REQ-CE-AUTHZ-002	REQ-CE-AUTHZ-003
T-CE-DATA-DISC	REQ-CE-CONF-001 REQ-CE-DECOM-001 REQ-CE-DECOM-002	REQ-CE-CONF-002	REQ-CE-CONF-003
T-CE-AVAIL-DOS	REQ-CE-ISO-001	REQ-CE-ISO-002	REQ-CE-ISO-003
T-CE-LOG-INT	REQ-CE-AUTHZ-001	REQ-CE-AUTHZ-002	REQ-CE-AUTHZ-003
T-CE-UPD-COMP	REQ-CE-INTEG-001 REQ-CE-UPD-001	REQ-CE-INTEG-002 REQ-CE-UPD-002	REQ-CE-INTEG-003
T-CE-CFG-COMP	REQ-CE-ISO-001 REQ-CE-AUTHZ-001 REQ-CE-CONF-001	REQ-CE-ISO-002 REQ-CE-AUTHZ-002 REQ-CE-CONF-002	REQ-CE-ISO-003 REQ-CE-AUTHZ-003 REQ-CE-CONF-003
T-CE-SC-COMP	REQ-CE-INTEG-001 REQ-CE-UPD-001	REQ-CE-INTEG-002 REQ-CE-UPD-002	REQ-CE-INTEG-003

B.8.2.3 CO

Table B.8.2.3-1: CO - threats to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	**Low**	**Medium**	**High**
T-CO-INTF-ESC	REQ-CO-ISO-001	REQ-CO-ISO-002	REQ-CO-ISO-003
T-CO-INTEG-TRUST	REQ-CO-INTEG-001	REQ-CO-INTEG-002	REQ-CO-INTEG-003
T-CO-UNAUTH-ADMIN	REQ-CO-AUTH-001	REQ-CO-AUTH-002	REQ-CO-AUTH-003
T-CO-UNAUTH-FUNC	REQ-CO-AUTHZ-001	REQ-CO-AUTHZ-002	REQ-CO-AUTHZ-003
T-CO-DATA-DISC	REQ-CO-CONF-001 REQ-CO-DECOM-001 REQ-CO-DECOM-002	REQ-CO-CONF-002	REQ-CO-CONF-003
T-CO-AVAIL-DOS	REQ-CO-ISO-001	REQ-CO-ISO-002	REQ-CO-ISO-003
T-CO-LOG-INT	REQ-CO-AUTHZ-001	REQ-CO-AUTHZ-002	REQ-CO-AUTHZ-003
T-CO-UPD-COMP	REQ-CO-UPD-001	REQ-CO-UPD-002	REQ-CO-UPD-003
T-CO-CFG-COMP	REQ-CO-INTEG-001 REQ-CO-ISO-001 REQ-CO-AUTHZ-001	REQ-CO-INTEG-002 REQ-CO-ISO-002 REQ-CO-AUTHZ-002	REQ-CO-INTEG-003 REQ-CO-ISO-003 REQ-CO-AUTHZ-003

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
	REQ-CO-CONF-001	REQ-CO-CONF-002	REQ-CO-CONF-003
T-CO-SC-COMP	REQ-CO-INTEG-001 REQ-CO-UPD-001	REQ-CO-INTEG-002 REQ-CO-UPD-002	REQ-CO-INTEG-003 REQ-CO-UPD-003

B.8.3 Hardware-level common threat

Table B.8.3-1: Hypervisor - HW-level threat to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-ALL-HW-ACCESS	REQ-H-IMG-INT-001 REQ-H-CONF-001	REQ-H-B-INT-002 REQ-H-IMG-INT-002 REQ-H-RP-INT-002 REQ-H-CONF-002 REQ-H-CONF-003	REQ-H-B-INT-002 REQ-H-IMG-INT-003 REQ-H-RP-INT-002 REQ-H-RA-INT-003, REQ-H-RA-INT-004 REQ-H-CONF-004

Table B.8.3-2: CRS - HW-level threat to requirements mapping

Threat ID	Basic Requirements	Elevated Requirements	Advanced Requirements
Associated Risk Level	Low	Medium	High
T-ALL-HW-ACCESS	REQ-CRS-IMG-INT-001 REQ-CRS-CONF-001	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-002 REQ-CRS-RP-INT-002 REQ-CRS-CONF-002 REQ-CRS-CONF-003	REQ-CRS-B-INT-002 REQ-CRS-IMG-INT-003 REQ-CRS-RP-INT-002 REQ-CRS-RA-INT-003, REQ-CRS-RA-INT-004 REQ-CRS-CONF-004

Annexes C to J:
Void

Annex K (normative): Generic cryptographic requirements and assessment

K.1 Cryptography

K.1.1 Requirement

The product's default configuration shall only use cryptographic mechanisms that meet at least one of the following criteria:

1) ACM-listed: the cryptographic mechanism is listed in the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [1];

2) ACM-extended: the cryptographic mechanism is not listed in the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [1] and meets at least one of the following conditions:

   a. the cryptographic mechanism is listed in clause K.3.2 as an ACM-extended cryptographic mechanism for the specific product function(s);

   b. where the cryptographic mechanism is not listed in clause K.3.2, the cryptographic mechanism meets all the following criteria:

      i) the cryptographic mechanism, or where applicable the ACM-listed cryptographic mechanism on which it is based, is not deprecated per the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [1];

      ii) the cryptographic mechanism has been specified, developed or maintained through a transparent process by a recognised European, international or sector-specific standards development organisation, or by an industry specification organisation accountable for the relevant specification, including ISO, IEC, IETF, IEEE, ETSI, CEN and CENELEC*;* or the cryptographic mechanism is listed as suitable in a publicly available cryptographic catalogue maintained by a recognised national or governmental cybersecurity authority, where the catalogue is maintained under a documented revision and retirement process, including NIST FIPS and Special Publications, NIST CAVP/ACVP validation programmes, BSI TR-02102 series [i.14], [i.15], [i.16], [i.17], ANSSI RGS Annex B [i.18] and ETSI TS 119 312 [16];

      iii) the cryptographic mechanism is described in a valid, publicly available and uniquely referenceable specification;

      iv) the cryptographic properties of the cryptographic mechanism are known;

      v) no known weakness affects the cryptographic mechanism in a way that affects its cryptographic properties;

      vi) the cryptographic mechanism is required for a specific set of product functions;

3) Interoperability-based: the cryptographic mechanism is listed in clause K.4.2 as an interoperability-based cryptographic mechanism for specific product function(s) and external specification(s) or external requirement(s).

NOTE 1: The reference to the product's default configuration is intended to define a clear and assessable baseline, corresponding to the configuration in which the product is placed on the market. The product can provide several configurations that fulfil the requirement.

NOTE 2: For products supplied as hardware platforms or components to be configured by an integrator, references in this annex to the product's default configuration refer to the configuration specified for the intended operational use of the product after integration, including the relevant integration assumptions and configuration constraints.

NOTE 3: The applicable ACM catalogue version is identified in the normative references of the present document. The lifecycle treatment of mechanisms affected by ACM deprecation dates, expiry dates, migration conditions or usage limitations is addressed in clause K.2.

NOTE 4: In this clause, an external specification or external requirement means a specification or requirement that is imposed on the product, and which requires the use of a specific cryptographic mechanism for the product to interoperate with an identified system, platform or operational context. An external requirement can be a regulatory requirement, operational constraint, technical interoperability constraint, or platform compatibility requirement.

NOTE 5: Inclusion of a cryptographic mechanism under the interoperability-based criterion does not classify that mechanism as state-of-the-art cryptography.

EXAMPLE: Examples of product functions include secure communication based on TLS 1.3; authenticated encryption of communicated data based on AES-GCM; storage confidentiality based on AES-XTS; firmware signature verification based on Ed25519; and key derivation based on HKDF.

K.1.2 Assessment of product cryptographic configuration

K.1.2.0 General

The assessment in clause K.1.2 verifies the cryptographic mechanisms used in the product's default configuration against clause K.1.1.

For ACM-extended cryptographic mechanisms, the assessment verifies that the cryptographic mechanism is either listed in clause K.3.2 or fulfils the criteria specified in clause K.1.1 item 2.b. For interoperability-based cryptographic mechanisms, the assessment verifies that the cryptographic mechanism is listed in clause K.4.2. The assessment also verifies that the product uses the cryptographic mechanism in accordance with the relevant characteristics, product functions, use cases where applicable, external specifications or external requirements, and conditions specified in the present document.

K.1.2.1 Assessment of ACM-listed cryptographic mechanisms

K.1.2.1.1 Assessment objective

The purpose of this assessment case is to verify that cryptographic mechanisms claimed to comply with clause K.1.1 item 1) and used in the product's default configuration are listed in the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [1].

K.1.2.1.2 Assessment preparation

- Preconditions for the assessment: The product's default configuration shall be used for the assessment.
- The documentation shall provide a list of cryptographic mechanisms used in the product's default configuration and claimed under clause K.1.1 item 1), including the related product function, relevant parameters where applicable, and the relevant ACM entry.

K.1.2.1.3 Assessment activities

The assessment shall include verification that:

a) each cryptographic mechanism claimed under clause K.1.1 item 1) is identified by reference to a relevant ACM entry;

b) the ACM entry corresponds to the cryptographic mechanism used in the product's default configuration, including relevant parameters where applicable;

c) where the ACM entry includes lifecycle information, such as expiry date, deprecation date, migration condition or usage limitation, this information is reflected in the documentation.

K.1.2.1.4 Assessment evidence

The assessment evidence shall include, as applicable:

a) description of the product's default configuration;

b) list of cryptographic mechanisms claimed under clause K.1.1 item 1);

c) references to the relevant ACM entries;

d) relevant parameters, profiles, cipher suites or configuration constraints, where applicable;

e) relevant lifecycle information from the ACM catalogue, where applicable.

K.1.2.1.5 Assessment verdict

- The verdict PASS shall be assigned if the required evidence has been provided, is complete, is applicable to the assessed configuration, and demonstrates compliance with clause K.1.1 item 1).
- The verdict FAIL shall be assigned otherwise.

K.1.2.2 Assessment of ACM-extended cryptographic mechanisms

K.1.2.2.1 Assessment objective

The purpose of this assessment case is to verify that cryptographic mechanisms claimed to comply with clause K.1.1 item 2) and used in the product's default configuration are either listed as ACM-extended cryptographic mechanisms in clause K.3.2 or fulfil the criteria specified in clause K.1.1 item 2)b, and are used for the claimed product function(s).

K.1.2.2.2 Assessment preparation

- Preconditions for the assessment: The product's default configuration shall be used for the assessment.
- The documentation shall provide a list of cryptographic mechanisms used in the product's default configuration and claimed under clause K.1.1 item 2), including the related product function, use case where applicable, relevant parameters where applicable, and at least one of the following:

  a) the relevant entry in clause K.3.2; or

  b) evidence that the cryptographic mechanism fulfils all applicable criteria in clause K.1.1 item 2.b.

K.1.2.2.3 Assessment activities

The assessment shall include verification that:

a) each cryptographic mechanism claimed under clause K.1.1 item 2) is either listed in clause K.3.2 as an ACM-extended cryptographic mechanism or supported by evidence demonstrating fulfilment of the applicable criteria in clause K.1.1 item 2.b;

b) the cryptographic mechanism used by the product corresponds to the mechanism listed in clause K.3.2 or described in the evidence provided under clause K.1.1 item 2.b;

c) the product function using the cryptographic mechanism, and the use case where applicable, correspond to the related product function and use case specified in clause K.3.2 or justified under clause K.1.1 item 2.b;

d) the relevant characteristics, parameters, profiles, cipher suites or configuration constraints of the cryptographic mechanism correspond to those specified in clause K.3.2 or justified under clause K.1.1 item 2.b, where applicable;

e) where clause K.3.2 or the justification under clause K.1.1 item 2.b defines conditions or limitations for the use of the cryptographic mechanism, these conditions or limitations are reflected in the product's default configuration;

f) where technically feasible, the cryptographic mechanism, parameters and configuration identified in the documentation correspond to the assessed product configuration.

K.1.2.2.4 Assessment evidence

The assessment evidence shall include, as applicable:

a) description of the product's default configuration;

b) list of cryptographic mechanisms claimed under clause K.1.1 item 2);

c) reference to the relevant entry in clause K.3.2, where the mechanism is listed in clause K.3.2;

d) evidence that the cryptographic mechanism fulfils the criteria in clause K.1.1 item 2.b, where compliance with clause K.1.1 item 2 is claimed on the basis of those criteria;

e) identification of the related product function using the cryptographic mechanism and the use case where applicable;

f) relevant characteristics, parameters, profiles, cipher suites or configuration constraints, where applicable;

g) evidence that the cryptographic mechanism is used in accordance with the conditions or limitations specified in clause K.3.2 or in the justification under clause K.1.1 item 2.b, where applicable.

K.1.2.2.5 Assessment verdict

- The verdict PASS shall be assigned if the required evidence has been provided, is complete, is applicable to the assessed configuration, and demonstrates compliance with clause K.1.1 item 2).
- The verdict FAIL shall be assigned otherwise.

K.1.2.3 Assessment of interoperability-based cryptographic mechanisms

K.1.2.3.1 Assessment objective

The purpose of this assessment case is to verify that cryptographic mechanisms claimed to comply with clause K.1.1 item 3) and used in the product's default configuration are listed as interoperability-based cryptographic mechanisms in clause K.4.2 and are used in accordance with the characteristics, product functions, use cases where applicable, external specifications or external requirements, and conditions specified in clause K.4.2.

K.1.2.3.2 Assessment preparation

- Preconditions for the assessment: The product's default configuration shall be used for the assessment.
- The documentation shall provide a list of cryptographic mechanisms used in the product's default configuration and claimed under clause K.1.1 item 3), including the related product function, use case where applicable, relevant parameters where applicable, the external specification or external requirement requiring its use, and the relevant entry in clause K.4.2.

K.1.2.3.3 Assessment activities

The assessment shall include verification that:

a) each cryptographic mechanism claimed under clause K.1.1 item 3) is listed in clause K.4.2 as an interoperability-based cryptographic mechanism;

b) the cryptographic mechanism used by the product corresponds to the cryptographic mechanism listed in clause K.4.2;

c) the product function using the cryptographic mechanism, and the use case where applicable, correspond to the related product function and use case specified in clause K.4.2;

d) the external specification or external requirement requiring the use of the cryptographic mechanism corresponds to the external specification or external requirement specified in clause K.4.2;

e) the relevant characteristics, parameters, profiles, cipher suites or configuration constraints of the cryptographic mechanism correspond to those specified in clause K.4.2, where applicable;

f) the conditions or limitations specified in clause K.4.2 for the use of the cryptographic mechanism are reflected in the product's default configuration;

g) where technically feasible, the cryptographic mechanism, parameters and configuration identified in the documentation correspond to the assessed product configuration.

K.1.2.3.4 Assessment evidence

The assessment evidence shall include, as applicable:

a) description of the product's default configuration;

b) list of cryptographic mechanisms claimed under clause K.1.1 item 3);

c) reference to the relevant entry in clause K.4.2;

d) identification of the related product function using the cryptographic mechanism and the use case where applicable;

e) identification of the external specification or external requirement requiring use of the cryptographic mechanism;

f) relevant characteristics, parameters, profiles, cipher suites or configuration constraints, where applicable;

g) evidence that the cryptographic mechanism is used in accordance with the conditions or limitations specified in clause K.4.2.

K.1.2.3.5 Assessment verdict

- The verdict PASS shall be assigned if the required evidence has been provided, is complete, is applicable to the assessed configuration, and demonstrates compliance with clause K.1.1 item 3).
- The verdict FAIL shall be assigned otherwise.

NOTE 1: The assessment of the external specification or external requirement itself is outside the scope of this assessment case. The assessment verifies that the external specification or external requirement is identified in clause K.4.2 and that the cryptographic mechanism is used by the product for the related product function.

NOTE 2: Documentation review is a minimum assessment activity under the present annex. Functional and/or resistance testing activities can be added by vertical standards where relevant for the product category, product function or cryptographic mechanism.

K.2 Crypto agility

K.2.1 Requirement

Where the product's default configuration uses a cryptographic mechanism for which the ACM catalogue, or the present document, specifies a deprecation date, expiry date, migration condition or usage limitation falling within the intended lifetime of the product, the product shall provide means for addressing the affected cryptographic mechanism by one or more of the following:

a) updating the cryptographic mechanism;

b) using another cryptographic mechanism that complies with clause K.1.1 and is not subject to the relevant deprecation date, expiry date, migration condition or usage limitation;

c) disabling the use of the affected cryptographic mechanism; or

d) limiting the use of the affected product functions accordingly.

EXAMPLE: Where a security mechanism uses a hybrid cryptographic construction, for example a hybrid key-encapsulation mechanism combining a classical and a post-quantum primitive, the lifecycle status of each constituent primitive is relevant to the lifecycle status of the construction. Hybridization can be used as part of a planned migration strategy where permitted by the ACM catalogue or by the present document. However, hybridization does not, by itself, extend the recommended usage lifetime of a constituent primitive that has been deprecated.

NOTE 1: Where a hybrid cryptographic construction includes multiple cryptographic primitives, the lifecycle status of each constituent primitive is relevant to the lifecycle status of the construction. Hybridization is not assumed to mitigate the deprecation of a constituent primitive unless the ACM catalogue or the present document classifies the hybrid construction as suitable for the corresponding product function.

NOTE 2: Crypto agility supports maintaining appropriate cryptographic protection within the intended lifetime of the product, in addition to the capability of updating cryptographic mechanisms on the product in accordance with secure update and secure communication mechanisms.

NOTE 3: Deprecation information from sources other than the ACM catalogue or the present document can be considered as input to vulnerability management or security risk assessment but does not by itself trigger the requirement in clause K.2.1.

NOTE 4: Where the product provides cryptographic capabilities for use by other products, components or layers, the mechanism required by clause K.2.1 can consist of update, configuration, disabling, deprecation or limitation capabilities usable by the integrating product, system operator or user, where applicable.

NOTE 5: The applicability condition of this requirement can express exceptions based on product characteristics, e.g. cryptographic implementation based on immutable hardware co-processor(s) or hardware-based root of trust, or long-lived silicon used in a long-lived non-updateable environment.

K.2.2 Assessment of crypto-agility

K.2.2.1 Assessment objective

The purpose of this assessment case is to verify whether, where the product's default configuration uses a cryptographic mechanism for which the ACM catalogue, or the present document, specifies a deprecation date, expiry date, migration condition or usage limitation falling within the intended lifetime of the product, the product provides means to address the affected cryptographic mechanism in accordance with clause K.2.1.

K.2.2.2 Assessment preparation

- Preconditions for the assessment: The product's default configuration shall be used for the assessment.
- The documentation shall identify:

  a) the intended lifetime of the product;

  b) the cryptographic mechanisms used in the product's default configuration;

  c) the related product function(s) for each cryptographic mechanism;

  d) the lifecycle information applicable to each cryptographic mechanism, where specified by the ACM catalogue or the present document, including any deprecation date, expiry date, migration condition or usage limitation.

K.2.2.3 Assessment activities

The assessment shall include verification that:

a) for each cryptographic mechanism used in the product's default configuration, the lifecycle information specified by the ACM catalogue or the present document has been identified, where such information is specified;

b) where the ACM catalogue or the present document specifies a deprecation date, expiry date, migration condition or usage limitation for a cryptographic mechanism, this information is reflected in the documentation;

c) where a deprecation date, expiry date, migration condition or usage limitation falls within the intended lifetime of the product, the product provides an applicable mechanism for updating the cryptographic mechanism, using another cryptographic mechanism that complies with clause K.1.1 and is not subject to the relevant deprecation date, expiry date, migration condition or usage limitation, disabling the use of the affected cryptographic mechanism, or limiting the use of the affected product functions accordingly;

d) where another cryptographic mechanism is used, that cryptographic mechanism complies with clause K.1.1 and is not subject to the relevant deprecation date, expiry date, migration condition or usage limitation.

K.2.2.4 Assessment evidence

The assessment evidence shall include, as applicable:

a) documentation of the intended lifetime of the product;

b) list of cryptographic mechanisms used in the product's default configuration;

c) identification of the related product function(s) for each cryptographic mechanism;

d) references to the ACM catalogue entry or to the provision of the present document used to determine lifecycle information, where applicable;

e) identification of any deprecation date, expiry date, migration condition or usage limitation falling within the intended lifetime of the product;

f) description of the means provided by the product, such as update of the cryptographic mechanism, use of another cryptographic mechanism that complies with clause K.1.1 and is not subject to the relevant lifecycle constraint, disabling the use of the affected cryptographic mechanism, or limitation of the use of the affected product functions.

K.2.2.5 Assessment verdict

- The verdict PASS shall be assigned if the required evidence has been provided, is complete, is applicable to the assessed configuration, and demonstrates compliance with clause K.2.1.
- The verdict FAIL shall be assigned otherwise.

K.3 ACM-extended cryptographic mechanisms

K.3.1 Requirement

Where the product's default configuration uses ACM-extended cryptographic mechanisms, these mechanisms shall comply with the ACM-extended criterion in clause K.1.1 item 2.

K.3.2 List of ACM-extended cryptographic mechanisms

K.3.2.1 Stream cipher mechanisms

K.3.2.1.1 ChaCha20

ChaCha20, as specified in IRTF RFC 8439 [2], may be used as a state-of-the-art stream cipher for CES/VES products where it is used within a protocol profile or construction that provides the required security properties for the intended use.

Table K.3.2.1.1-1: ChaCha20

Mechanism	Public specification	Security properties	Intended CES/VES use	Conditions
ChaCha20	IRTF RFC 8439 [2]	Confidentiality, when used with unique nonces and appropriate key management	Protected communication and software-based encryption contexts, especially where AES hardware acceleration is not available	ChaCha20 shall not be used alone where integrity or authenticity is required. Where communication or data protection requires integrity, ChaCha20 shall be used as part of an authenticated construction or combined with an appropriate integrity mechanism.

K.3.2.2 Authenticated encryption mechanisms

K.3.2.2.1 ChaCha20-Poly1305

ChaCha20-Poly1305, as specified in IRTF RFC 8439 [2], may be used as a state-of-the-art authenticated encryption mechanism for CES/VES products where AEAD protection is required and where the mechanism is used within a suitable protocol or implementation profile.

Table K.3.2.2.1-1: ChaCha20-Poly1305

Mechanism	Public specification	Security properties	Intended CES/VES use	Conditions
ChaCha20-Poly1305	IRTF RFC 8439 [2]	Confidentiality, integrity and data-origin authentication for protected data	Protected communication channels, API protection, control-plane communication, service-to-service communication, secure metadata exchange, and protected software-based communication where AES acceleration is not available	Nonces shall be generated and managed according to the specification and the applicable protocol profile. Reuse of a nonce with the same key shall be prevented.

K.3.2.3 Message authentication mechanisms

K.3.2.3.1 Poly1305

Poly1305, as specified in IRTF RFC 8439 [2], may be used as a state-of-the-art message authentication mechanism only where it is used according to a specification or protocol profile that ensures correct one-time key use.

Table K.3.2.3.1-1: Poly1305

Mechanism	Public specification	Security properties	Intended CES/VES use	Conditions
Poly1305	IRTF RFC 8439 [2]	Message integrity and data-origin authentication	Primarily as part of ChaCha20-Poly1305 or another specified construction	Poly1305 shall not be used as an ad hoc standalone MAC unless the implementation ensures correct one-time key generation and use according to the applicable specification.

K.3.2.4 Password protection mechanisms

K.3.2.4.1 Argon2id

Argon2id, as specified in IRTF RFC 9106 [3], may be used as a state-of-the-art password hashing and password-based key derivation mechanism for CES/VES products where password verifiers are stored by the product or where encryption keys are derived from passphrases.

Table K.3.2.4.1-1: Argon2id

Mechanism	Public specification	Security properties	Intended CES/VES use	Conditions
Argon2id	IRTF RFC 9106 [3]	Resistance against offline password guessing through memory-hard password hashing	Protection of password verifiers for local administrative accounts, emergency accounts, product-managed users, service accounts, or registry credentials stored by the product; and derivation of encryption keys from passphrases where the product implements passphrase-protected storage encryption, such as full-disk or volume encryption of product data or configuration storage.	Each password verifier shall use a unique random salt of at least 128 bits. Where Argon2id is used as a key derivation function for storage encryption, the derived key length shall meet the requirements of the encryption mechanism, and the salt shall not be reused across volumes or derived keys. Parameters shall be selected according to the product threat model and deployment constraints and shall provide resistance to offline guessing attacks. Parameters shall be documented and justified.

K.3.2.5 Elliptic-curve key agreement mechanisms

K.3.2.5.1 X25519 and X448

X25519 and X448, as specified in IRTF RFC 7748 [5], may be used as state-of-the-art elliptic-curve Diffie-Hellman key agreement mechanisms where they are supported by the applicable protocol profile and where the resulting shared secret is used only through a suitable key derivation function or protocol-defined key schedule.

Table K.3.2.5.1-1: X25519 and X448

Mechanism	Public specification	Security properties	Intended CES/VES use	Conditions
X25519	IRTF RFC 7748 [5]; IETF RFC 8410 [6] for X.509 algorithm identifiers	Key agreement	TLS, SSH, internal secure-channel establishment, service-to-service communication, workload identity systems, and protected management channels where supported by the protocol profile	The shared secret shall be processed through a suitable key derivation function or protocol-defined key schedule. The mechanism shall be used only in protocol profiles that define peer authentication and protection against active attacks where required.
X448	IRTF RFC 7748 [5]; IETF RFC 8410 [6] for X.509 algorithm identifiers	Key agreement at a higher classical security level	Same as X25519 where a higher classical security level is required and supported	Same as X25519.

NOTE: X25519 and X448 provide key agreement, not authentication by themselves. Authentication depends on the surrounding protocol, certificates, signatures, pre-shared credentials, or other authenticated key-exchange framework.

K.3.2.6 Digital signature mechanisms

K.3.2.6.1 Ed25519 and Ed448

Ed25519 and Ed448, as specified in IRTF RFC 8032 [7], may be used as state-of-the-art digital signature mechanisms for CES/VES products where supported by the applicable protocol, certificate profile, artefact-signing framework, update framework, or trust framework.

Table K.3.2.6.1-1: Ed25519 and Ed448

Mechanism	Public specification	Security properties	Intended CES/VES use	Conditions
Ed25519	IRTF RFC 8032 [7]; IETF RFC 8410 [6] for X.509 algorithm identifiers	Integrity and authenticity of signed artefacts; accountability support where the trust and key-management model supports it	Container image signatures, VM image signatures, update signatures, orchestration artefact signatures, configuration signatures, SSH keys, attestation artefacts, and certificate-based authentication where supported	The trust model, key-management model, certificate or key-distribution model, revocation model, and signature verification policy shall be documented.
Ed448	IRTF RFC 8032 [7]; IETF RFC 8410 [6] for X.509 algorithm identifiers	Integrity and authenticity of signed artefacts at a higher classical security level; accountability support where the trust and key-management model supports it	Same as Ed25519 where a higher classical security level is required and supported	Same as Ed25519.

NOTE: NIST FIPS 186-5 [i.12] specifies EdDSA as a digital signature algorithm.

K.3.2.7 Container image, artefact, attestation and metadata protection formats

K.3.2.7.1 COSE

COSE, as specified in IETF RFC 9052 [8] and IETF RFC 9053 [9], may be used as a state-of-the-art protection format for CBOR-based security artefacts in CES/VES products.

Table K.3.2.7.1-1: COSE

Mechanism	Public specification	Security properties	Intended CES/VES use	Conditions
COSE	IETF RFC 9052 [8]; IETF RFC 9053 [9]	Depends on the selected COSE algorithm: signature, MAC, encryption, or authenticated encryption	Attestation evidence, compact claims, signed metadata, signed configuration, provenance records, and CBOR-based security artefacts	This entry covers COSE only as a CBOR-based message syntax and algorithm-binding framework. For each intended CES/VES use, the COSE structure, COSE algorithm identifier(s), key type, parameters and trust model shall be identified.

K.3.2.7.2 JOSE / JWS / JWE

JOSE, including JWS and JWE, as specified in IETF RFC 7515 [10] and IETF RFC 7516 [11], may be used as a state-of-the-art protection format for JSON-based security artefacts in CES/VES products.

Table K.3.2.7.2-1: JOSE / JWS / JWE

Mechanism	Public specification	Security properties	Intended CES/VES use	Conditions
JOSE / JWS / JWE	IETF RFC 7515 [10]; IETF RFC 7516 [11]; related JOSE specifications	Depends on the selected JOSE algorithm: signature, encryption, or authenticated encryption	Tokens, signed claims, signed metadata, configuration artefacts, provenance records, workload identity assertions, API security artefacts	This entry covers JOSE/JWS/JWE only as a JSON-based syntax and algorithm-binding framework. For each intended CES/VES use, the JOSE object type, selected alg value and, where applicable, enc value, key type, parameters and trust model shall be identified.

NOTE: COSE, JOSE, JWS and JWE are protection formats. They are included in this annex because they are commonly used in CES/VES ecosystems for artefact protection, token protection, metadata protection, and attestation-related claims. Their inclusion does not approve every algorithm registered for use with those formats.

K.3.2.8 Protected communication profiles

K.3.2.8.1 QUIC

QUIC version 1, as specified in IETF RFC 9000 [12] and secured using TLS as specified in IETF RFC 9001 [13], may be used as a state-of-the-art protected communication protocol for CES/VES products where the product exposes or consumes QUIC-based interfaces.

Table K.3.2.8.1-1: QUIC

Protocol	Public specification	Security properties	Intended CES/VES use	Conditions
QUIC version 1	IETF RFC 9000 [12]; IETF RFC 9001 [13]	Confidentiality, integrity and peer authentication as provided by the TLS 1.3 profile used with QUIC	API communication, control-plane communication, telemetry, management or orchestration communication where QUIC is implemented	This entry covers QUIC version 1 only when used with TLS 1.3 as specified in IETF RFC 9001 [13].

K.3.2.9 Secure time protocols

K.3.2.9.1 Network Time Security for NTP

Network Time Security for NTP, as specified in IRTF RFC 8915 [14], may be used as a state-of-the-art secure time mechanism where CES/VES products rely on NTP for security-relevant time.

Table K.3.2.9.1-1: NTS for NTP

Protocol	Public specification	Security properties	Intended CES/VES use	Conditions
Network Time Security for NTP	IRTF RFC 8915 [14]	Authentication and integrity protection for time synchronization, depending on the deployment model	Certificate validation, audit-log time-stamping, attestation freshness, event ordering, forensic evidence, update validation, and security monitoring where NTP time is security-relevant	This entry covers NTS for NTP where the product relies on NTP time for security-relevant functions. NTS-KE shall use TLS 1.3. The NTS/TLS profile, certificate validation policy, AEAD algorithm for NTP packet protection, key lengths, cookie handling and key refresh behaviour shall be identified for the intended CES/VES use. NTP without NTS is outside the scope of this entry for security-relevant time synchronization.

K.3.2.10 Secure device and platform identity

K.3.2.10.1 IEEE 802.1AR secure device identity

IEEE 802.1AR [15] secure device identity, as specified in IEEE 802.1AR [15], may be used as a state-of-the-art device or platform identity mechanism where CES/VES products rely on platform-bound identities, node identities, host identities, or device identities.

Table K.3.2.10.1-1: IEEE 802.1AR [15] secure device identity

Mechanism	Public specification	Security properties	Intended CES/VES use	Conditions
IEEE 802.1AR [15] secure device identity	IEEE 802.1AR [15]	Device identity, platform identity, certificate-based identity binding	Hypervisor host identity, CRS node identity, CO node identity, M&O-managed node identity, attestation identity, service identity bootstrapping	This entry covers IEEE 802.1AR [15] secure device identity for certificate-based device or platform identity binding. For each intended CES/VES use, the DevID type, certificate profile, signature algorithm, public-key algorithm, key size, certificate validation policy, revocation handling, trust-anchor management and private-key protection mechanism shall be identified.

K.3.3 Assessment

Compliance with the ACM-extended criterion in clause K.1.1 item 2) is assessed in clause K.1.2.2.

K.4 Interoperability-based cryptographic mechanisms

K.4.1 Requirement

Where the product's default configuration uses interoperability-based cryptographic mechanisms, these mechanisms shall be selected from the interoperability-based cryptographic mechanisms listed in clause K.4.2 for the specified product functions and external specifications or external requirements and shall be used in accordance with the conditions or limitations specified in clause K.4.2.

K.4.2 List of interoperability-based cryptographic mechanisms

K.4.2.1 Password protection mechanisms

K.4.2.1.1 scrypt

scrypt, as specified in IETF RFC 7914 [4], may be used as an interoperability-based password-based key derivation or password hashing mechanism where its use is required by an identified external specification, platform compatibility requirement, operational constraint, technical interoperability constraint, stored password-verifier format, encrypted storage format, or established deployment profile.

Table K.4.2.1.1-1: scrypt

Mechanism	Public specification	Security properties	Intended CES/VES use	Interoperability or compatibility justification	Conditions
scrypt	IETF RFC 7914 [4]	Resistance against offline password guessing through memory-hard password-based key derivation	Protection of password verifiers or derivation of keys from passwords in CES/VES products	Required by an identified external specification, platform compatibility requirement, operational constraint, technical interoperability constraint, stored password-verifier format, encrypted storage format, or established deployment profile	Use shall be limited to the identified interoperability or compatibility purpose. Each password verifier shall use a unique random salt of at least 128 bits. Parameters N, r and p shall be selected and justified according to the product threat model, deployment constraints and intended lifetime. For new product-controlled password-verifier formats or password-based key-derivation formats, Argon2id should be preferred unless the rationale for selecting scrypt is documented.

K.4.3 Assessment

Compliance with the requirement in clause K.4.1 is assessed in clause K.1.2.3.

NOTE: The assessment of the external specifications or external requirements themselves is outside the scope of this assessment. Their relevance is deemed confirmed by their inclusion in the present document.

Annexes L to Q:
Void

Annex R (normative): Additional provisions for products relying on Remote Data Processing Solutions (RDPS)

R.1 Scope & Applicability

This annex specifies additional technical requirements and assessment provisions for products with digital elements that rely on Remote Data Processing Solutions ("RDPS") for the provision or support of one or more product functions.

The requirements applicable to the product functions themselves remain specified in the core part of the present document and continue to apply to the product as a whole, including where a product function is performed locally, supported by an RDPS, or performed in part through an RDPS.

This annex specifies supplementary requirements and assessment provisions for the product-facing RDPS boundary, i.e. for the additional attack surface, interactions, and dependency conditions introduced where a product function is performed or supported across the local product side and the RDPS side.

Where the product does not rely on remote data processing solutions for the provision or support of any product function, this annex is not applicable.

This annex does not extend to the full remote service environment or underlying third-party infrastructure, except to the extent necessary to define assumptions, dependencies, and assessment conditions relevant to the product-facing RDPS boundary and the secure operation of the RDPS-dependent product function.

Where an M&O System, CE, or CO is part of the declared VES or CES product, that component is assessed under the applicable core requirements of the present document. Such a component is not treated as an RDPS for the purpose of Annex R solely because it is remotely deployed, distributed, or accessed remotely.

Annex R applies to a remote data processing solution that is distinct from the M&O System, CE, or CO component already covered as part of the declared VES or CES product, and that provides or supports one or more RDPS-dependent product functions across a product-facing RDPS boundary.

Where the RDPS side is part of the declared product, provided by the manufacturer, or under the responsibility of the manufacturer for the RDPS-dependent product function, the applicable RDPS-side requirements of this annex apply.

Where the RDPS side is outside the declared product boundary and outside the responsibility of the manufacturer, the assessment identifies the corresponding Operational Environment dependency, the expected security properties, and available third-party assurance evidence, where relevant.

R.2 RDPS as a product-boundary extension

When a product relies on RDPS for the provision or support of one or more product functions, the use of RDPS is considered, for the purposes of the present annex, to introduce an additional product-facing boundary and interface between the local product side and the RDPS side.

The present annex does not restate or replace the requirements applicable to the product function as such. It specifies only the supplementary requirements needed to address the increased attack surface introduced by the RDPS boundary.

The present annex therefore addresses the security implications introduced by that additional boundary and interface, including:

a) the exchanged interactions across the RDPS boundary;

b) the relevant product-side and RDPS-side endpoints; and

c) the dependency conditions arising where the secure operation of the product function relies on interactions across that boundary.

The scope of the present annex is limited to the RDPS interface between the local product side and the RDPS side.

For the purposes of the present annex:

- local product side means the product elements within the product boundary that participate in the interaction with the RDPS;
- RDPS means the remote data processing solution performing or supporting operations relevant to the RDPS-dependent product function;
- RDPS interface means the communication boundary between the product-side endpoint and the RDPS-side endpoint.

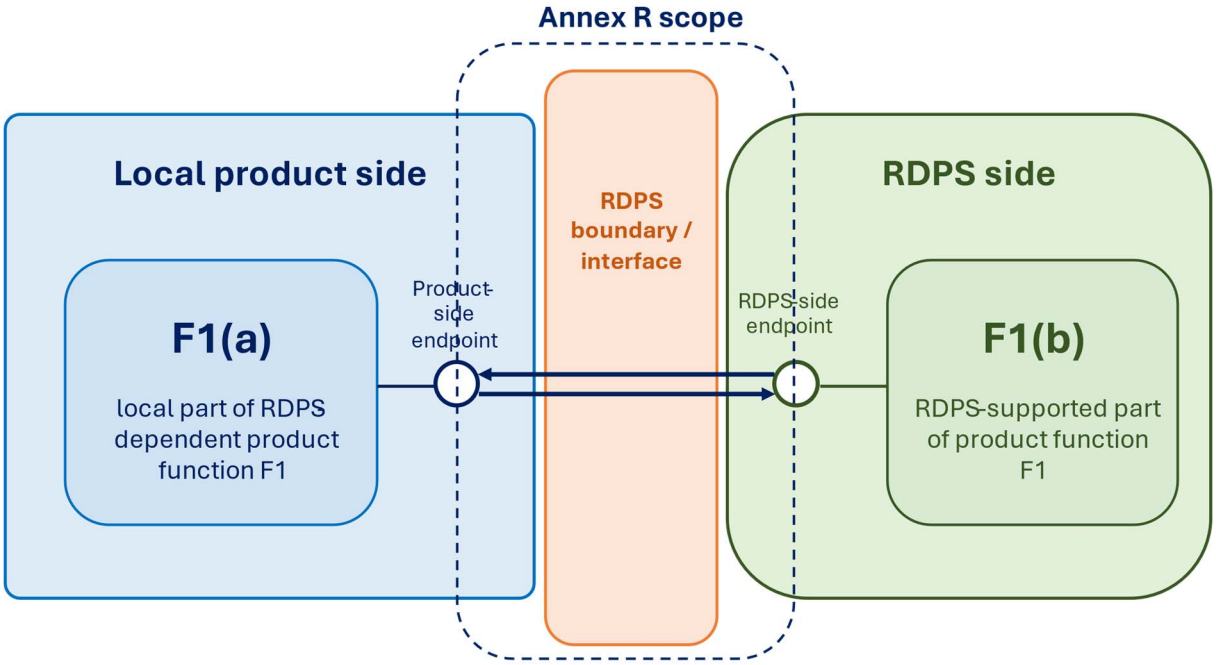

Figure R.2-1: RDPS model for Annex R - scope focused on the product ↔ RDPS boundary

R.3 Threat Model

R.3.0 General

This clause does not restate or replace threats for local product components that are already defined in the main clauses of the present document. Instead, it focuses on:

- threats introduced by the local product ↔ RDPS interface; and
- threats introduced by the RDPS as a remote service capability.

The threat model:

- identifies RDPS assets; and
- defines a set of threats applicable to RDPS.

R.3.1 Assets

The threat model considers the following assets:

DA-RDPS-001 - RDPS interaction content: The information exchanged across the RDPS boundary, including, as applicable, requests, commands, responses, decisions, configuration data, secrets, update-related data, and personal data where such data is exchanged.

For the purposes of DA-RDPS-001, Table R.3.1-1 identifies potential categories of RDPS interaction content and indicates the security properties to be considered for each category when determining the applicable Annex R requirement families.

Table R.3.1-1: Security properties for DA-RDPS-001

Content type	Data authenticity	Integrity	Confidentiality
Commands / requests & Responses / decisions	X	X	
Configuration data	X	X	
Secrets (e.g. keys)	X	X	X
Code or update data	X	X	
Confidential code or update data	X	X	X
Personal data		X	X

NOTE: In Table R.3.1-1, "data authenticity" refers to the authenticity or trusted origin of the exchanged interaction content. It does not refer to authentication of the RDPS boundary endpoint.

DA-RDPS-002 – RDPS-dependent product function: product function for which one or more operations depend on an RDPS, including the associated security-relevant state, decisions, outputs, or configuration.

For the purposes of DA-RDPS-002, Table R.3.2-1 identifies the security property relevant to the RDPS-dependent product function.

Table R.3.2-1: Security property for DA-RDPS-002

Content type	Availability
RDPS-dependent product function	X

R.3.2 Threat catalogue

The following threats constitute the baseline threat set considered for the RDPS boundary within the scope of Annex R, covering the RDPS interaction content exchanged across the boundary and the availability of the RDPS-dependent product function.

T-RDPS-01 Loss of data authenticity of interactions across the RDPS boundary

An attacker causes an endpoint to accept interaction content across the RDPS boundary whose claimed source or origin is not genuine.

Threat type: Spoofing / impersonation

Security impact: The receiving endpoint may rely on interaction content that does not originate from the intended source, which can affect the behavior, state, configuration, or security of the RDPS-dependent product function.

T-RDPS-02 Loss of integrity of interactions across the RDPS boundary

An interaction exchanged across the RDPS boundary is modified, substituted, injected, corrupted, or replayed before it is processed by the receiving endpoint.

Threat type: Tampering

Security impact: The receiving endpoint may act upon interaction content whose contents no longer correspond to what was sent or intended by the legitimate source, which can affect the behavior, state, configuration, or security of the RDPS-dependent product function.

T-RDPS-03 Unauthorized disclosure of interactions across the RDPS boundary

Data exchanged across the RDPS boundary is disclosed to an unauthorized party.

Threat type: Information disclosure

Security impact: An attacker may obtain interaction content whose disclosure can affect the security or secure operation of the RDPS-dependent product function.

T-RDPS-04 Unavailability, unacceptable delay, or intolerable degradation of interactions across the RDPS boundary

The local product side or the RDPS side is unable to send, receive, or process interactions across the RDPS boundary, or receives them only after an unacceptable delay, or experiences degradation of communication performance across the RDPS boundary to an extent that is not tolerable for the successful operation of the RDPS-dependent product function.

Threat type: Denial of service

Security impact: The RDPS-dependent product function may fail to operate as intended, may operate with incomplete, outdated, or degraded information, may remain blocked, or may continue without the required interaction being available when needed.

R.3.3 Assets ↔ Threats mapping

Table R.3.3-1: Threats ↔ Assets mapping

Threat ID	Threat title	Related asset(s)
T-RDPS-01	Loss of data authenticity of interactions across the RDPS boundary	DA-RDPS-001
T-RDPS-02	Loss of integrity of interactions across the RDPS boundary	DA-RDPS-001
T-RDPS-03	Unauthorized disclosure of interactions across the RDPS boundary	DA-RDPS-001
T-RDPS-04	Unavailability, unacceptable delay, or intolerable degradation of interactions across the RDPS boundary	DA-RDPS-002

R.4 Security Requirements

R.4.1 General

The requirements in the present clause are organized by requirement family. The requirement families address the primary security properties identified for the assets in clause R.3.1, namely data authenticity, integrity, confidentiality, and availability.

Within each requirement family, the requirements are expressed as a sequence of three requirements with increasing rigour. The lower-numbered requirement defines the baseline security outcome to be achieved, while the higher-numbered requirements define strengthened requirements for cases where increased protection is needed.

The applicable requirement level is selected according to the Security Profile applicable to the RDPS-dependent product function. SP1 corresponds to the -001 requirements, SP2 corresponds to the -002 requirements, and SP3 corresponds to the -003 requirements.

The requirements within a given requirement family represent alternative levels of rigour and are not cumulative. For each applicable requirement family, one applicable level is selected. Selection of a higher-numbered requirement within a family includes the security outcome of the lower-numbered requirement or requirements in that family and does not require separate selection of those lower-numbered requirements.

R.4.2 Applicability of Annex R requirements

The requirements in the present clause address the protection of the security properties identified for the assets in clause R.3.1.

Where a security property is identified as relevant for a given asset in the asset tables of clause R.3.1, the corresponding requirement family or families in the present clause shall be considered for applicability to that asset.

For DA-RDPS-001, the primary requirement families considered for applicability are:

a) data authenticity of interactions;

b) integrity of interactions; and

c) confidentiality of exchanged data.

For DA-RDPS-002, the primary requirement family considered for applicability is availability.

R.4.3 Local product side requirements

R.4.3.1 Data authenticity of interactions received from the RDPS side

REQ-RDPS-L-AUTH-001

The local product side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, that the interaction originates from the intended RDPS-side endpoint.

Rationale: This requirement protects against acceptance of interaction content whose claimed source or origin is not genuine.

REQ-RDPS-L-AUTH-002

The local product side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, the data authenticity of that interaction using cryptographic endpoint authentication, an authenticated communication channel, or an equivalent cryptographic mechanism that provides assurance of the origin of the interaction.

Rationale: This requirement strengthens protection against spoofed or falsely originated interaction content by requiring cryptographic assurance of origin.

REQ-RDPS-L-AUTH-003

The local product side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, that:

a) the cryptographic authenticity of the interaction is bound to the intended RDPS-side endpoint; and

b) replayed or stale interactions are detected and rejected where replay or reuse could affect the secure operation of the RDPS-dependent product function.

Rationale: This requirement provides stronger protection by requiring source-bound authenticity and replay protection.

R.4.3.2 Integrity of interactions received from the RDPS side

REQ-RDPS-L-INTEG-001

The local product side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, that the interaction has not been modified, substituted, corrupted, or injected.

Rationale: This requirement protects against loss of integrity of received interaction content.

REQ-RDPS-L-INTEG-002

The local product side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, the integrity of that interaction using a cryptographic integrity-protection mechanism.

Rationale: This requirement strengthens protection against unauthorized modification or substitution of received interactions.

REQ-RDPS-L-INTEG-003

The local product side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, that the interaction is protected by a cryptographic integrity-protection mechanism bound to the authenticated source of the interaction.

Rationale: This requirement provides stronger protection by requiring cryptographic integrity protection bound to the authenticated source.

R.4.3.3 Confidentiality of data exchanged with the RDPS side

REQ-RDPS-L-CONF-001

Where disclosure of data exchanged with the RDPS side could lead to unacceptable cybersecurity risk for the RDPS-dependent product function, the local product side shall protect the confidentiality of that data during exchange with the RDPS side.

Rationale: Confidentiality is not required for every exchange. However, where disclosure of data exchanged with the RDPS side could affect the security or secure operation of the RDPS-dependent product function, confidentiality protection is required.

REQ-RDPS-L-CONF-002

Where disclosure of data exchanged with the RDPS side could lead to unacceptable cybersecurity risk for the RDPS-dependent product function, the local product side shall protect the confidentiality of that data during exchange with the RDPS side using cryptographic confidentiality protection.

Rationale: This requirement strengthens confidentiality protection by requiring cryptographic protection of exchanged data where disclosure would create unacceptable cybersecurity risk.

REQ-RDPS-L-CONF-003

Where disclosure of data exchanged with the RDPS side could lead to unacceptable cybersecurity risk for the RDPS-dependent product function, the local product side shall use cryptographic confidentiality protection that is established with the intended RDPS-side endpoint and that prevents disclosure of the exchanged data to endpoints other than that intended endpoint.

Rationale: This requirement provides stronger protection by ensuring that confidential exchanged data is disclosed only to the intended peer endpoint.

R.4.3.4 Availability

REQ-RDPS-L-AVAIL-001

Where the availability or timeliness of interactions received from the RDPS side is necessary for the secure operation of the RDPS-dependent product function, the local product side shall:

a) detect unavailability of the RDPS side, unacceptable delay of expected interactions, or intolerable degradation of the interaction;

b) use timeout or equivalent timeliness controls for expected RDPS-side interactions; and

c) apply a defined behaviour for the RDPS-dependent product function.

Rationale: This requirement ensures that loss, delay, or degradation of RDPS-side interactions is detected and handled in a defined manner before it adversely affects the secure operation of the RDPS-dependent product function.

REQ-RDPS-L-AVAIL-002

Where the availability or timeliness of interactions received from the RDPS side is necessary for the secure operation of the RDPS-dependent product function, the local product side shall:

a) detect unavailability of the RDPS side, unacceptable delay of expected interactions, or intolerable degradation of the interaction;

b) use timeout or equivalent timeliness controls for expected RDPS-side interactions;

c) apply a defined degraded behaviour or secure state when the required RDPS-side interactions are unavailable, delayed beyond the acceptable time, or degraded beyond acceptable limits; and

d) support recovery of the RDPS-dependent product function using locally maintained state, configuration, or other recovery information where such information is necessary to restore secure operation.

Rationale: This requirement strengthens protection by requiring controlled degraded operation or transition to a secure state, together with support for restoration of secure operation.

REQ-RDPS-L-AVAIL-003

Where the availability or timeliness of interactions received from the RDPS side is necessary for the secure operation of the RDPS-dependent product function, the local product side shall:

a) detect unavailability of the RDPS side, unacceptable delay of expected interactions, or intolerable degradation of the interaction;

b) use timeout or equivalent timeliness controls for expected RDPS-side interactions;

c) apply a defined degraded behaviour or secure state when the required RDPS-side interactions are unavailable, delayed beyond the acceptable time, or degraded beyond acceptable limits;

d) support recovery of the RDPS-dependent product function using locally maintained state, configuration, or other recovery information where such information is necessary to restore secure operation;

e) support continuity of operation through failover, redundancy, or an equivalent resilience mechanism where such a mechanism is necessary.

Rationale: This requirement provides stronger protection by requiring resilience measures.

R.4.4 RDPS side requirements

R.4.4.1 Data authenticity of interactions received from the local product side

REQ-RDPS-R-AUTH-001

The RDPS side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, that the interaction originates from the intended local product-side endpoint.

Rationale: This requirement protects against acceptance of interaction content whose claimed source or origin is not genuine.

REQ-RDPS-R-AUTH-002

The RDPS side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, the data authenticity of that interaction using cryptographic endpoint authentication, an authenticated communication channel, or an equivalent cryptographic mechanism that provides assurance of the origin of the interaction.

Rationale: This requirement strengthens protection against spoofed or falsely originated interaction content by requiring cryptographic assurance of origin.

REQ-RDPS-R-AUTH-003

The RDPS side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, that:

a) the cryptographic authenticity of the interaction is bound to the intended local product-side endpoint; and

b) replayed or stale interactions are detected and rejected where replay or reuse could affect the secure operation of the RDPS-dependent product function.

Rationale: This requirement provides stronger protection by requiring source-bound authenticity and replay protection.

R.4.4.2 Integrity of interactions received from the local product side

REQ-RDPS-R-INTEG-001

The RDPS side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, that the interaction has not been modified, substituted, corrupted, or injected.

Rationale: This requirement protects against loss of integrity of received interaction content.

REQ-RDPS-R-INTEG-002

The RDPS side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, the integrity of that interaction using a cryptographic integrity-protection mechanism.

Rationale: This requirement strengthens protection against unauthorized modification or substitution of received interactions.

REQ-RDPS-R-INTEG-003

The RDPS side shall verify, before acting upon a received interaction that can influence the behaviour, state, configuration, or security of the RDPS-dependent product function, that the interaction is protected by a cryptographic integrity-protection mechanism bound to the authenticated source of the interaction.

Rationale: This requirement provides stronger protection by requiring cryptographic integrity protection bound to the authenticated source.

R.4.4.3 Confidentiality of data exchanged with the local product side

REQ-RDPS-R-CONF-001

Where disclosure of data exchanged with the local product side could lead to unacceptable cybersecurity risk for the RDPS-dependent product function, the RDPS side shall protect the confidentiality of that data during exchange with the local product side.

Rationale: This requirement ensures confidentiality where disclosure of exchanged data could affect the security or secure operation of the RDPS-dependent product function.

REQ-RDPS-R-CONF-002

Where disclosure of data exchanged with the local product side could lead to unacceptable cybersecurity risk for the RDPS-dependent product function, the RDPS side shall protect the confidentiality of that data during exchange with the local product side using cryptographic confidentiality protection.

Rationale: This requirement strengthens confidentiality protection by requiring cryptographic protection during exchange.

REQ-RDPS-R-CONF-003

Where disclosure of data exchanged with the local product side could lead to unacceptable cybersecurity risk for the RDPS-dependent product function, the RDPS side shall use cryptographic confidentiality protection that is established with the intended local product-side endpoint and that prevents disclosure of the exchanged data to endpoints other than that intended endpoint.

Rationale: This requirement provides stronger protection by ensuring that confidential exchanged data is disclosed only to the intended peer endpoint.

R.4.4.4 Availability

REQ-RDPS-R-AVAIL-001

Where the availability or timeliness of interactions received from the local product side is necessary for the secure operation of the RDPS-dependent product function, the RDPS side shall:

a) detect unavailability of the local product side, unacceptable delay of expected interactions, or intolerable degradation of the interaction;

b) use timeout or equivalent timeliness controls for expected local product-side interactions; and

c) apply a defined behaviour for the RDPS-dependent product function.

Rationale: This requirement ensures that loss, delay, or degradation of local product-side interactions is detected and handled in a defined manner before it adversely affects the secure operation of the RDPS-dependent product function.

REQ-RDPS-R-AVAIL-002

Where the availability or timeliness of interactions received from the local product side is necessary for the secure operation of the RDPS-dependent product function, the RDPS side shall:

a) detect unavailability of the local product side, unacceptable delay of expected interactions, or intolerable degradation of the interaction;

b) use timeout or equivalent timeliness controls for expected local product-side interactions;

c) apply a defined degraded behaviour or secure state when the required local product-side interactions are unavailable, delayed beyond the acceptable time, or degraded beyond acceptable limits; and

d) support recovery of the RDPS-dependent product function using maintained state, configuration, or other recovery information where such information is necessary to restore secure operation.

Rationale: This requirement strengthens protection by requiring controlled degraded operation or transition to a secure state, together with support for restoration of secure operation.

REQ-RDPS-R-AVAIL-003

Where the availability or timeliness of interactions received from the local product side is necessary for the secure operation of the RDPS-dependent product function, the RDPS side shall:

a) detect unavailability of the local product side, unacceptable delay of expected interactions, or intolerable degradation of the interaction;

b) use timeout or equivalent timeliness controls for expected local product-side interactions;

c) apply a defined degraded behaviour or secure state when the required local product-side interactions are unavailable, delayed beyond the acceptable time, or degraded beyond acceptable limits;

d) support recovery of the RDPS-dependent product function using maintained state, configuration, or other recovery information where such information is necessary to restore secure operation; and

e) support continuity of operation through failover, redundancy, or an equivalent resilience mechanism where such a mechanism is necessary.

Rationale: This requirement provides stronger protection by requiring resilience measures.

R.4.5 Threats ↔ Requirements mapping

Table R.4.5-1: Threats ↔ Requirements mapping

Threat ID	Requirement(s) - Local product side	Requirement(s) - RDPS side
T-RDPS-01	REQ-RDPS-L-AUTH-001 REQ-RDPS-L-AUTH-002 REQ-RDPS-L-AUTH-003	REQ-RDPS-R-AUTH-001 REQ-RDPS-R-AUTH-002 REQ-RDPS-R-AUTH-003
T-RDPS-02	REQ-RDPS-L-INTEG-001 REQ-RDPS-L-INTEG-002 REQ-RDPS-L-INTEG-003	REQ-RDPS-R-INTEG-001 REQ-RDPS-R-INTEG-002 REQ-RDPS-R-INTEG-003
T-RDPS-03	REQ-RDPS-L-CONF-001 REQ-RDPS-L-CONF-002 REQ-RDPS-L-CONF-003	REQ-RDPS-R-CONF-001 REQ-RDPS-R-CONF-002 REQ-RDPS-R-CONF-003
T-RDPS-04	REQ-RDPS-L-AVAIL-001 REQ-RDPS-L-AVAIL-002 REQ-RDPS-L-AVAIL-003	REQ-RDPS-R-AVAIL-001 REQ-RDPS-R-AVAIL-002 REQ-RDPS-R-AVAIL-003

R.5 Conformity assessment

R.5.0 General

The assessment cases in the present clause are aligned with the requirements in clause R.4. Each assessment case follows a common structure: assessment reference, assessment objective, assessment preparation, assessment activities, assessment verdict, and assessment evidence.

Where the RDPS side is part of the declared product, provided by the manufacturer, or under the responsibility of the manufacturer for the RDPS-dependent product function, the applicable RDPS-side assessment cases in clause R.5.2 are assessed using evidence for that RDPS side.

Where the RDPS side is outside the declared product boundary and outside the responsibility of the manufacturer, the applicable assessment identifies the corresponding Operational Environment dependency, the expected security properties, and available third-party assurance evidence, where relevant. Such evidence may include design documentation, contractual or service documentation, configuration guidance, security attestations, certification evidence, audit reports, or other assurance evidence relevant to the RDPS boundary.

R.5.1 Local product side requirements

R.5.1.1 Assessment case for REQ-RDPS-L-AUTH-001

1. Assessment Reference
Requirement: REQ-RDPS-L-AUTH-001

2. Assessment objective
The objective of the assessment is to verify that the local product side verifies, before acting upon a relevant received interaction, that the interaction originates from the intended RDPS-side endpoint.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the interactions received from the RDPS side that can influence the behaviour, state, configuration, or security of that function;
- the intended RDPS-side endpoint for those interactions; and
- the mechanism used by the local product side to determine the origin of the received interaction.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the mechanism used by the local product side to determine the origin of the relevant interaction;
- verify that the origin of the interaction is checked before the interaction is acted upon;
- verify that interactions from the intended RDPS-side endpoint are accepted; and
- verify that interactions not originating from the intended RDPS-side endpoint are rejected or not acted upon.

5. Assessment verdict
Pass:

- the local product side verifies the origin of the relevant interaction before acting upon it; and
- interactions from endpoints other than the intended RDPS-side endpoint are rejected or not acted upon.

Fail:

- the local product side acts upon a relevant interaction without verifying its origin; or
- the local product side accepts and acts upon a relevant interaction not originating from the intended RDPS-side endpoint.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design or configuration evidence describing how origin of the interaction is determined;
- evidence showing acceptance of interactions from the intended RDPS-side endpoint; and
- evidence showing rejection or non-processing of interactions from other endpoints.

R.5.1.2 Assessment case for REQ-RDPS-L-AUTH-002

1. Assessment Reference
Requirement: REQ-RDPS-L-AUTH-002

2. Assessment objective
The objective of the assessment is to verify that the local product side verifies the data authenticity of relevant received interactions using cryptographic endpoint authentication, an authenticated communication channel, or an equivalent cryptographic mechanism providing assurance of origin.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions;
- the cryptographic mechanism used to provide assurance of origin; and
- the conditions under which the local product side accepts or rejects those interactions.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the cryptographic mechanism used to provide assurance of origin;
- verify that the local product side checks the authenticity-protection status of the interaction before acting upon it;
- verify that a valid interaction protected by the intended mechanism is accepted; and

- verify that an interaction lacking valid cryptographic assurance of origin is rejected or not acted upon.

5. Assessment verdict
Pass:

- the local product side uses a cryptographic mechanism providing assurance of origin for relevant received interactions;
- the local product side verifies that protection before acting upon the interaction; and
- interactions without valid assurance of origin are rejected or not acted upon.

Fail:

- the local product side acts upon a relevant interaction without verifying cryptographic assurance of origin; or
- the local product side accepts and acts upon an interaction for which the required authenticity mechanism is absent or invalid.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for the cryptographic authenticity mechanism;
- traces or logs showing successful acceptance of valid protected interactions; and
- traces or logs showing rejection or non-processing of interactions without valid assurance of origin.

R.5.1.3 Assessment case for REQ-RDPS-L-AUTH-003

1. Assessment Reference
Requirement: REQ-RDPS-L-AUTH-003

2. Assessment objective
The objective of the assessment is to verify that the local product side verifies that the cryptographic authenticity of the received interaction is bound to the intended RDPS-side endpoint and that replayed or stale interactions are detected and rejected where replay or reuse could affect secure operation.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions;
- the mechanism used to bind authenticity to the intended RDPS-side endpoint; and
- the replay or freshness protection mechanism, where applicable.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the mechanism that binds the authenticity of the interaction to the intended RDPS-side endpoint;
- inspect the design and configuration of the freshness, anti-replay, or equivalent mechanism used;
- verify that a valid interaction bound to the intended RDPS-side endpoint is accepted;
- verify that an interaction with valid-looking authenticity protection but not bound to the intended RDPS-side endpoint is rejected or not acted upon; and
- where replay or reuse could affect secure operation, verify that replayed or stale interactions are detected and rejected.

5. Assessment verdict
Pass:

- the local product side verifies that authenticity is bound to the intended RDPS-side endpoint; and
- replayed or stale interactions are rejected where replay or reuse could affect secure operation.

Fail:

- the local product side accepts and acts upon an interaction whose cryptographic authenticity is not bound to the intended RDPS-side endpoint; or
- the local product side fails to detect and reject replayed or stale interactions where such protection is required.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for source binding and anti-replay/freshness controls;
- evidence showing acceptance of valid bound interactions; and
- evidence showing rejection of misbound, replayed, or stale interactions.

R.5.1.4 Assessment case for REQ-RDPS-L-INTEG-001

1. Assessment Reference
Requirement: REQ-RDPS-L-INTEG-001

2. Assessment objective
The objective of the assessment is to verify that the local product side checks that a relevant received interaction has not been modified, substituted, corrupted, or injected before acting upon it.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions; and
- the mechanism used by the local product side to detect modification, substitution, corruption, or injection.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the integrity-checking mechanism;
- verify that integrity is checked before the interaction is acted upon;
- verify that an unmodified valid interaction is accepted; and
- verify that modified, substituted, corrupted, or injected interactions are rejected or not acted upon.

5. Assessment verdict
Pass:

- the local product side checks the integrity of relevant received interactions before acting upon them; and
- modified, substituted, corrupted, or injected interactions are rejected or not acted upon.

Fail:

- the local product side acts upon a relevant interaction without verifying its integrity; or
- the local product side accepts and acts upon an altered, substituted, corrupted, or injected interaction.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design or configuration evidence of the integrity-checking mechanism;
- evidence showing acceptance of valid interactions; and
- evidence showing rejection or non-processing of altered or injected interactions.

R.5.1.5 Assessment case for REQ-RDPS-L-INTEG-002

1. Assessment Reference
Requirement: REQ-RDPS-L-INTEG-002

2. Assessment objective
The objective of the assessment is to verify that the local product side verifies the integrity of relevant received interactions using a cryptographic integrity-protection mechanism.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions; and
- the cryptographic integrity-protection mechanism used for those interactions.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the cryptographic integrity-protection mechanism;
- verify that integrity verification takes place before the interaction is acted upon;
- verify that a valid cryptographically protected interaction is accepted; and
- verify that an interaction with missing or invalid integrity protection is rejected or not acted upon.

5. Assessment verdict
Pass:

- the local product side uses cryptographic integrity protection for relevant received interactions;
- integrity verification is performed before the interaction is acted upon; and
- interactions with invalid or missing integrity protection are rejected or not acted upon.

Fail:

- the local product side does not use cryptographic integrity verification where required; or
- the local product side accepts and acts upon an interaction whose integrity protection is missing or invalid.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for the cryptographic integrity mechanism;
- evidence showing acceptance of valid protected interactions; and
- evidence showing rejection of interactions with invalid or missing integrity protection.

R.5.1.6 Assessment case for REQ-RDPS-L-INTEG-003

1. Assessment Reference
Requirement: REQ-RDPS-L-INTEG-003

2. Assessment objective
The objective of the assessment is to verify that the local product side verifies that the relevant received interaction is protected by a cryptographic integrity-protection mechanism bound to the authenticated source of the interaction.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions;
- the authenticated source of those interactions; and
- the mechanism used to bind integrity protection to that authenticated source.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the mechanism that binds integrity protection to the authenticated source;
- verify that a valid interaction whose integrity protection is correctly bound to the authenticated source is accepted;
- verify that an interaction with integrity protection not bound to the authenticated source is rejected or not acted upon; and
- verify that integrity verification and source binding checks are performed before the interaction is acted upon.

5. Assessment verdict
Pass:

- the local product side verifies that the interaction is protected by cryptographic integrity protection bound to the authenticated source; and
- interactions not correctly bound to that source are rejected or not acted upon.

Fail:

- the local product side accepts and acts upon a relevant interaction whose integrity protection is not bound to the authenticated source; or
- the local product side acts upon the interaction before performing the required checks.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for integrity binding to authenticated source;
- evidence showing acceptance of correctly bound interactions; and
- evidence showing rejection of misbound interactions.

R.5.1.7 Assessment case for REQ-RDPS-L-CONF-001

1. Assessment Reference
Requirement: REQ-RDPS-L-CONF-001

2. Assessment objective
The objective of the assessment is to verify that, where disclosure of exchanged data could lead to unacceptable cybersecurity risk for the RDPS-dependent product function, the local product side protects the confidentiality of that data during exchange with the RDPS side.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the categories of exchanged data for which disclosure could lead to unacceptable cybersecurity risk; and
- the mechanism used by the local product side to protect confidentiality during exchange.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the confidentiality-protection mechanism;
- verify that the identified data is protected during exchange with the RDPS side; and
- verify that data designated as requiring confidentiality is not exchanged in an unprotected manner.

5. Assessment verdict
Pass:

- the local product side protects the confidentiality of exchanged data where disclosure could lead to unacceptable cybersecurity risk; and
- such data is not exchanged in an unprotected manner.

Fail:

- the local product side does not protect the confidentiality of exchanged data where such protection is required; or
- identified confidential data is exchanged without appropriate confidentiality protection.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design or configuration evidence identifying the protected data and the means of protection; and
- evidence showing that relevant exchanged data is protected during exchange.

R.5.1.8 Assessment case for REQ-RDPS-L-CONF-002

1. Assessment Reference
Requirement: REQ-RDPS-L-CONF-002

2. Assessment objective
The objective of the assessment is to verify that the local product side protects the confidentiality of relevant exchanged data using cryptographic confidentiality protection.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the exchanged data requiring confidentiality protection; and
- the cryptographic confidentiality-protection mechanism used during exchange.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the cryptographic confidentiality-protection mechanism;
- verify that the identified exchanged data is protected by that mechanism during exchange; and
- verify that exchange of such data without the required cryptographic confidentiality protection is prevented.

5. Assessment verdict
Pass:

- relevant exchanged data is protected using cryptographic confidentiality protection; and
- exchange without the required cryptographic protection is prevented.

Fail:

- the local product side exchanges relevant confidential data without cryptographic confidentiality protection; or
- the configured confidentiality mechanism is not effectively applied to the relevant data exchange.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for the cryptographic confidentiality mechanism; and
- evidence showing protected exchange of the relevant data.

R.5.1.9 Assessment case for REQ-RDPS-L-CONF-003

1. Assessment Reference
Requirement: REQ-RDPS-L-CONF-003

2. Assessment objective
The objective of the assessment is to verify that the local product side uses cryptographic confidentiality protection established with the intended RDPS-side endpoint and preventing disclosure of exchanged data to endpoints other than that intended endpoint.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the exchanged data requiring stronger confidentiality protection;
- the intended RDPS-side endpoint; and
- the cryptographic mechanism used to establish confidentiality protection with that endpoint.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the cryptographic confidentiality mechanism and its association with the intended RDPS-side endpoint;
- verify that protected exchange is established with the intended RDPS-side endpoint;
- verify that the local product side does not disclose the protected data to other endpoints under the same exchange context; and
- verify that exchanges not established with the intended RDPS-side endpoint are rejected or do not result in disclosure of the protected data.

5. Assessment verdict
Pass:

- confidentiality protection is established with the intended RDPS-side endpoint;
- protected data is disclosed only to that intended endpoint; and
- exchanges with other endpoints do not result in disclosure of the protected data.

Fail:

- the local product side discloses protected data without establishing confidentiality protection with the intended RDPS-side endpoint; or
- the local product side discloses the protected data to endpoints other than the intended RDPS-side endpoint.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for endpoint-associated confidentiality protection;
- evidence showing protected disclosure to the intended RDPS-side endpoint; and
- evidence showing prevention of disclosure to other endpoints.

R.5.1.10 Assessment case for REQ-RDPS-L-AVAIL-001

1. Assessment Reference
Requirement: REQ-RDPS-L-AVAIL-001

2. Assessment objective
The objective of the assessment is to verify that the local product side detects unavailability, unacceptable delay, or intolerable degradation of required RDPS-side interactions, uses timeout or equivalent timeliness controls, and applies a defined behaviour for the RDPS-dependent product function.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the required RDPS-side interactions;
- the acceptable timing or service conditions for those interactions; and
- the defined behaviour to be applied when the required interactions are unavailable, delayed, or degraded.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the timeout, timeliness, or equivalent detection controls;
- verify that unavailability, unacceptable delay, or intolerable degradation of required interactions is detected;
- verify that a defined behaviour is applied when such conditions occur; and
- verify that, under normal conditions, expected interactions are processed without triggering the fault-handling behaviour.

5. Assessment verdict
Pass:

- the local product side detects the specified failure, delay, or degradation conditions;
- timeout or equivalent timeliness controls are used; and

- a defined behaviour is applied when the identified conditions occur.

Fail:

- the local product side does not detect the relevant failure, delay, or degradation condition; or
- the required defined behaviour is not applied when such a condition occurs.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for detection and timeliness controls;
- evidence showing detection of failure, delay, or degradation; and
- evidence showing application of the defined behaviour.

R.5.1.11 Assessment case for REQ-RDPS-L-AVAIL-002

1. Assessment Reference
Requirement: REQ-RDPS-L-AVAIL-002

2. Assessment objective
The objective of the assessment is to verify that the local product side applies a defined degraded behaviour or secure state when required RDPS-side interactions are unavailable, delayed, or degraded, and supports recovery of the RDPS-dependent product function where needed.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the required RDPS-side interactions;
- the degraded behaviour or secure state defined for failure conditions; and
- any locally maintained state, configuration, or other recovery information required to restore secure operation.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the degraded behaviour or secure-state mechanism;
- inspect the design and configuration of the recovery mechanism, where applicable;
- verify that when required interactions become unavailable, delayed beyond the acceptable time, or degraded beyond acceptable limits, the local product side applies the defined degraded behaviour or secure state; and
- verify that recovery support is available and can be used to restore secure operation where such support is required.

5. Assessment verdict
Pass:

- the local product side applies the defined degraded behaviour or secure state under the relevant failure conditions; and
- recovery support is available and usable where necessary to restore secure operation.

Fail:

- the local product side does not apply the defined degraded behaviour or secure state under the relevant failure conditions; or
- required recovery support is absent or ineffective.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for degraded behaviour, secure state, and recovery support; and
- evidence showing activation of degraded behaviour / secure state and restoration of secure operation where applicable.

R.5.1.12 Assessment case for REQ-RDPS-L-AVAIL-003

1. Assessment Reference
Requirement: REQ-RDPS-L-AVAIL-003

2. Assessment objective
The objective of the assessment is to verify that the local product side supports higher resilience, including continuity of operation through failover, redundancy, or equivalent resilience mechanisms where necessary.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the required RDPS-side interactions;
- the defined degraded behaviour or secure state;
- the recovery support; and
- the failover, redundancy, or equivalent resilience mechanism, where required.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the resilience mechanism used to support continuity of operation;
- verify that the defined degraded behaviour or secure state is applied under the relevant failure conditions;
- verify that recovery support is available where required;
- where failover, redundancy, or equivalent resilience is required, verify that continuity of operation is supported when the primary interaction path becomes unavailable, delayed, or degraded.

5. Assessment verdict
Pass:

- the local product side supports the required resilience mechanism where necessary;
- defined degraded behaviour or secure state is applied under the relevant conditions; and
- continuity of operation is supported in accordance with the design where resilience is required.

Fail:

- a required resilience mechanism is absent or ineffective; or
- the local product side fails to support continuity of operation where such support is required.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for resilience, degraded behaviour, and recovery support; and
- evidence showing failover, redundancy, or equivalent continuity support where required.

R.5.2 RDPS side requirements

R.5.2.1 Assessment case for REQ-RDPS-R-AUTH-001

1. Assessment Reference
Requirement: REQ-RDPS-R-AUTH-001

2. Assessment objective
The objective of the assessment is to verify that the RDPS side verifies, before acting upon a relevant received interaction, that the interaction originates from the intended local product-side endpoint.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the interactions received from the local product side that can influence the behaviour, state, configuration, or security of that function;
- the intended local product-side endpoint for those interactions; and
- the mechanism used by the RDPS side to determine the origin of the received interaction.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the mechanism used by the RDPS side to determine the origin of the relevant interaction;
- verify that the origin of the interaction is checked before the interaction is acted upon;
- verify that interactions from the intended local product-side endpoint are accepted; and
- verify that interactions not originating from the intended local product-side endpoint are rejected or not acted upon.

5. Assessment verdict
Pass:

- the RDPS side verifies the origin of the relevant interaction before acting upon it; and
- interactions from endpoints other than the intended local product-side endpoint are rejected or not acted upon.

Fail:

- the RDPS side acts upon a relevant interaction without verifying its origin; or
- the RDPS side accepts and acts upon a relevant interaction not originating from the intended local product-side endpoint.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design or configuration evidence describing how origin of the interaction is determined;
- evidence showing acceptance of interactions from the intended local product-side endpoint; and
- evidence showing rejection or non-processing of interactions from other endpoints.

R.5.2.2 Assessment case for REQ-RDPS-R-AUTH-002

1. Assessment Reference
Requirement: REQ-RDPS-R-AUTH-002

2. Assessment objective
The objective of the assessment is to verify that the RDPS side verifies the data authenticity of relevant received interactions using cryptographic endpoint authentication, an authenticated communication channel, or an equivalent cryptographic mechanism providing assurance of origin.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions;
- the cryptographic mechanism used to provide assurance of origin; and
- the conditions under which the RDPS side accepts or rejects those interactions.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the cryptographic mechanism used to provide assurance of origin;
- verify that the RDPS side checks the authenticity-protection status of the interaction before acting upon it;
- verify that a valid interaction protected by the intended mechanism is accepted; and
- verify that an interaction lacking valid cryptographic assurance of origin is rejected or not acted upon.

5. Assessment verdict
Pass:

- the RDPS side uses a cryptographic mechanism providing assurance of origin for relevant received interactions;
- the RDPS side verifies that protection before acting upon the interaction; and
- interactions without valid assurance of origin are rejected or not acted upon.

Fail:

- the RDPS side acts upon a relevant interaction without verifying cryptographic assurance of origin; or
- the RDPS side accepts and acts upon an interaction for which the required authenticity mechanism is absent or invalid.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for the cryptographic authenticity mechanism;
- traces or logs showing successful acceptance of valid protected interactions; and
- traces or logs showing rejection or non-processing of interactions without valid assurance of origin.

R.5.2.3 Assessment case for REQ-RDPS-R-AUTH-003

1. Assessment Reference
Requirement: REQ-RDPS-R-AUTH-003

2. Assessment objective
The objective of the assessment is to verify that the RDPS side verifies that the cryptographic authenticity of the received interaction is bound to the intended local product-side endpoint and that replayed or stale interactions are detected and rejected where replay or reuse could affect secure operation.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions;
- the mechanism used to bind authenticity to the intended local product-side endpoint; and
- the replay or freshness protection mechanism, where applicable.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the mechanism that binds the authenticity of the interaction to the intended local product-side endpoint;
- inspect the design and configuration of the freshness, anti-replay, or equivalent mechanism used;
- verify that a valid interaction bound to the intended local product-side endpoint is accepted;
- verify that an interaction with valid-looking authenticity protection but not bound to the intended local product-side endpoint is rejected or not acted upon; and
- where replay or reuse could affect secure operation, verify that replayed or stale interactions are detected and rejected.

5. Assessment verdict
Pass:

- the RDPS side verifies that authenticity is bound to the intended local product-side endpoint; and
- replayed or stale interactions are rejected where replay or reuse could affect secure operation.

Fail:

- the RDPS side accepts and acts upon an interaction whose cryptographic authenticity is not bound to the intended local product-side endpoint; or
- the RDPS side fails to detect and reject replayed or stale interactions where such protection is required.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for source binding and anti-replay/freshness controls;
- evidence showing acceptance of valid bound interactions; and
- evidence showing rejection of misbound, replayed, or stale interactions.

R.5.2.4 Assessment case for REQ-RDPS-R-INTEG-001

1. Assessment Reference
Requirement: REQ-RDPS-R-INTEG-001

2. Assessment objective
The objective of the assessment is to verify that the RDPS side checks that a relevant received interaction has not been modified, substituted, corrupted, or injected before acting upon it.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions; and

- the mechanism used by the RDPS side to detect modification, substitution, corruption, or injection.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the integrity-checking mechanism;
- verify that integrity is checked before the interaction is acted upon;
- verify that an unmodified valid interaction is accepted; and
- verify that modified, substituted, corrupted, or injected interactions are rejected or not acted upon.

5. Assessment verdict
Pass:

- the RDPS side checks the integrity of relevant received interactions before acting upon them; and
- modified, substituted, corrupted, or injected interactions are rejected or not acted upon.

Fail:

- the RDPS side acts upon a relevant interaction without verifying its integrity; or
- the RDPS side accepts and acts upon an altered, substituted, corrupted, or injected interaction.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design or configuration evidence of the integrity-checking mechanism;
- evidence showing acceptance of valid interactions; and
- evidence showing rejection or non-processing of altered or injected interactions.

R.5.2.5 Assessment case for REQ-RDPS-R-INTEG-002

1. Assessment Reference
Requirement: REQ-RDPS-R-INTEG-002

2. Assessment objective
The objective of the assessment is to verify that the RDPS side verifies the integrity of relevant received interactions using a cryptographic integrity-protection mechanism.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions; and
- the cryptographic integrity-protection mechanism used for those interactions.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the cryptographic integrity-protection mechanism;
- verify that integrity verification takes place before the interaction is acted upon;
- verify that a valid cryptographically protected interaction is accepted; and
- verify that an interaction with missing or invalid integrity protection is rejected or not acted upon.

5. Assessment verdict
Pass:

- the RDPS side uses cryptographic integrity protection for relevant received interactions;
- integrity verification is performed before the interaction is acted upon; and
- interactions with invalid or missing integrity protection are rejected or not acted upon.

Fail:

- the RDPS side does not use cryptographic integrity verification where required; or
- the RDPS side accepts and acts upon an interaction whose integrity protection is missing or invalid.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for the cryptographic integrity mechanism;
- evidence showing acceptance of valid protected interactions; and
- evidence showing rejection of interactions with invalid or missing integrity protection.

R.5.2.6 Assessment case for REQ-RDPS-R-INTEG-003

1. Assessment Reference
Requirement: REQ-RDPS-R-INTEG-003

2. Assessment objective
The objective of the assessment is to verify that the RDPS side verifies that the relevant received interaction is protected by a cryptographic integrity-protection mechanism bound to the authenticated source of the interaction.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the relevant received interactions;
- the authenticated source of those interactions; and
- the mechanism used to bind integrity protection to that authenticated source.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the mechanism that binds integrity protection to the authenticated source;
- verify that a valid interaction whose integrity protection is correctly bound to the authenticated source is accepted;
- verify that an interaction with integrity protection not bound to the authenticated source is rejected or not acted upon; and
- verify that integrity verification and source binding checks are performed before the interaction is acted upon.

5. Assessment verdict
Pass:

- the RDPS side verifies that the interaction is protected by cryptographic integrity protection bound to the authenticated source; and
- interactions not correctly bound to that source are rejected or not acted upon.

Fail:

- the RDPS side accepts and acts upon a relevant interaction whose integrity protection is not bound to the authenticated source; or
- the RDPS side acts upon the interaction before performing the required checks.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for integrity binding to authenticated source;
- evidence showing acceptance of correctly bound interactions; and
- evidence showing rejection of misbound interactions.

R.5.2.7 Assessment case for REQ-RDPS-R-CONF-001

1. Assessment Reference
Requirement: REQ-RDPS-R-CONF-001

2. Assessment objective
The objective of the assessment is to verify that, where disclosure of exchanged data could lead to unacceptable cybersecurity risk for the RDPS-dependent product function, the RDPS side protects the confidentiality of that data during exchange with the local product side.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the categories of exchanged data for which disclosure could lead to unacceptable cybersecurity risk; and
- the mechanism used by the RDPS side to protect confidentiality during exchange.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the confidentiality-protection mechanism;
- verify that the identified data is protected during exchange with the local product side; and
- verify that data designated as requiring confidentiality is not exchanged in an unprotected manner.

5. Assessment verdict
Pass:

- the RDPS side protects the confidentiality of exchanged data where disclosure could lead to unacceptable cybersecurity risk; and
- such data is not exchanged in an unprotected manner.

Fail:

- the RDPS side does not protect the confidentiality of exchanged data where such protection is required; or
- identified confidential data is exchanged without appropriate confidentiality protection.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design or configuration evidence identifying the protected data and the means of protection; and
- evidence showing that relevant exchanged data is protected during exchange.

R.5.2.8 Assessment case for REQ-RDPS-R-CONF-002

1. Assessment Reference
Requirement: REQ-RDPS-R-CONF-002

2. Assessment objective
The objective of the assessment is to verify that the RDPS side protects the confidentiality of relevant exchanged data using cryptographic confidentiality protection.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the exchanged data requiring confidentiality protection; and
- the cryptographic confidentiality-protection mechanism used during exchange.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the cryptographic confidentiality-protection mechanism;
- verify that the identified exchanged data is protected by that mechanism during exchange; and
- verify that exchange of such data without the required cryptographic confidentiality protection is prevented.

5. Assessment verdict
Pass:

- relevant exchanged data is protected using cryptographic confidentiality protection; and
- exchange without the required cryptographic protection is prevented.

Fail:

- the RDPS side exchanges relevant confidential data without cryptographic confidentiality protection; or
- the configured confidentiality mechanism is not effectively applied to the relevant data exchange.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for the cryptographic confidentiality mechanism; and
- evidence showing protected exchange of the relevant data.

R.5.2.9 Assessment case for REQ-RDPS-R-CONF-003

1. Assessment Reference
Requirement: REQ-RDPS-R-CONF-003

2. Assessment objective
The objective of the assessment is to verify that the RDPS side uses cryptographic confidentiality protection established with the intended local product-side endpoint and preventing disclosure of exchanged data to endpoints other than that intended endpoint.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the exchanged data requiring stronger confidentiality protection;
- the intended local product-side endpoint; and

- the cryptographic mechanism used to establish confidentiality protection with that endpoint.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the cryptographic confidentiality mechanism and its association with the intended local product-side endpoint;
- verify that protected exchange is established with the intended local product-side endpoint;
- verify that the RDPS side does not disclose the protected data to other endpoints under the same exchange context; and
- verify that exchanges not established with the intended local product-side endpoint are rejected or do not result in disclosure of the protected data.

5. Assessment verdict
Pass:

- confidentiality protection is established with the intended local product-side endpoint;
- protected data is disclosed only to that intended endpoint; and
- exchanges with other endpoints do not result in disclosure of the protected data.

Fail:

- the RDPS side discloses protected data without establishing confidentiality protection with the intended local product-side endpoint; or
- the RDPS side discloses the protected data to endpoints other than the intended local product-side endpoint.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for endpoint-associated confidentiality protection;
- evidence showing protected disclosure to the intended local product-side endpoint; and
- evidence showing prevention of disclosure to other endpoints.

R.5.2.10 Assessment case for REQ-RDPS-R-AVAIL-001

1. Assessment Reference
Requirement: REQ-RDPS-R-AVAIL-001

2. Assessment objective
The objective of the assessment is to verify that the RDPS side detects unavailability, unacceptable delay, or intolerable degradation of required local product-side interactions, uses timeout or equivalent timeliness controls, and applies a defined behaviour for the RDPS-dependent product function.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the required local product-side interactions;
- the acceptable timing or service conditions for those interactions; and
- the defined behaviour to be applied when the required interactions are unavailable, delayed, or degraded.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the timeout, timeliness, or equivalent detection controls;
- verify that unavailability, unacceptable delay, or intolerable degradation of required interactions is detected;
- verify that a defined behaviour is applied when such conditions occur; and
- verify that, under normal conditions, expected interactions are processed without triggering the fault-handling behaviour.

5. Assessment verdict
Pass:

- the RDPS side detects the specified failure, delay, or degradation conditions;
- timeout or equivalent timeliness controls are used; and
- a defined behaviour is applied when the identified conditions occur.

Fail:

- the RDPS side does not detect the relevant failure, delay, or degradation condition; or
- the required defined behaviour is not applied when such a condition occurs.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for detection and timeliness controls;
- evidence showing detection of failure, delay, or degradation; and
- evidence showing application of the defined behaviour.

R.5.2.11 Assessment case for REQ-RDPS-R-AVAIL-002

1. Assessment Reference
Requirement: REQ-RDPS-R-AVAIL-002

2. Assessment objective
The objective of the assessment is to verify that the RDPS side applies a defined degraded behaviour or secure state when required local product-side interactions are unavailable, delayed, or degraded, and supports recovery of the RDPS-dependent product function where needed.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the required local product-side interactions;
- the degraded behaviour or secure state defined for failure conditions; and
- any maintained state, configuration, or other recovery information required to restore secure operation.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the degraded behaviour or secure-state mechanism;
- inspect the design and configuration of the recovery mechanism, where applicable;

- verify that when required interactions become unavailable, delayed beyond the acceptable time, or degraded beyond acceptable limits, the RDPS side applies the defined degraded behaviour or secure state; and
- verify that recovery support is available and can be used to restore secure operation where such support is required.

5. Assessment verdict
Pass:

- the RDPS side applies the defined degraded behaviour or secure state under the relevant failure conditions; and
- recovery support is available and usable where necessary to restore secure operation.

Fail:

- the RDPS side does not apply the defined degraded behaviour or secure state under the relevant failure conditions; or
- required recovery support is absent or ineffective.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for degraded behaviour, secure state, and recovery support; and
- evidence showing activation of degraded behaviour / secure state and restoration of secure operation where applicable.

R.5.2.12 Assessment case for REQ-RDPS-R-AVAIL-003

1. Assessment Reference
Requirement: REQ-RDPS-R-AVAIL-003

2. Assessment objective
The objective of the assessment is to verify that the RDPS side supports higher resilience, including continuity of operation through failover, redundancy, or equivalent resilience mechanisms where necessary.

3. Assessment preparation
Before starting the assessment, the following shall be identified:

- the RDPS-dependent product function;
- the required local product-side interactions;
- the defined degraded behaviour or secure state;
- the recovery support; and
- the failover, redundancy, or equivalent resilience mechanism, where required.

4. Assessment activities
The assessment shall include the following activities:

- inspect the design and configuration of the resilience mechanism used to support continuity of operation;
- verify that the defined degraded behaviour or secure state is applied under the relevant failure conditions;
- verify that recovery support is available where required;
- where failover, redundancy, or equivalent resilience is required, verify that continuity of operation is supported when the primary interaction path becomes unavailable, delayed, or degraded.

5. Assessment verdict
Pass:

- the RDPS side supports the required resilience mechanism where necessary;
- defined degraded behaviour or secure state is applied under the relevant conditions; and
- continuity of operation is supported in accordance with the design where resilience is required.

Fail:

- a required resilience mechanism is absent or ineffective; or
- the RDPS side fails to support continuity of operation where such support is required.

6. Assessment evidence
The following evidence may be used to support the assessment:

- design and configuration evidence for resilience, degraded behaviour, and recovery support; and
- evidence showing failover, redundancy, or equivalent continuity support where required.

Annexes S to X:
Void

Annex Y (informative): Mapping of CES/VES cryptographic requirements to ACM and Annex K mechanism families

Y.1 General

This annex provides a mapping between requirements of the present document that explicitly require cryptographic functionality and the corresponding cryptographic mechanism families addressed through Annex K and the ECCG Agreed Cryptographic Mechanisms (ACM) catalogue [1].

Annex Y does not introduce additional requirements and does not modify the applicability, scope, or assurance expectations of clauses 5.2, 5.3 or Annex K. Compliance remains determined by the applicable requirements in clauses 5.2 and 5.3 and by the cryptographic requirements and assessment provisions in Annex K.

Y.2 Mapping of Hypervisor requirements

Table Y.2-1: Mapping of Hypervisor cryptographic requirements

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-H-CP-ISO-002	Cryptographically protected communication for network-accessible administrative or control-plane interfaces, providing confidentiality and integrity protection.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected as the protected communication profile.
REQ-H-CP-ISO-003	Mutually authenticated cryptographically protected communication for network-accessible administrative or control-plane interfaces.	ACM 6.1 TLS; ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-H-NP-ISO-003	Cryptographically protected management traffic where dedicated physical interfaces are not used, providing confidentiality, integrity and authentication.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-H-B-INT-002	Integrity and authenticity verification of Hypervisor executable artefacts before use, using signatures, certificates, digests, measurements, trusted reference values or equivalent metadata.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected for signatures. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used as signed metadata/protection formats.
REQ-H-IMG-INT-001	Integrity verification of guest VM images against a trusted reference value.	ACM 2.3 Hash Functions	Not applicable.
REQ-H-IMG-INT-002	Integrity and authenticity verification of guest VM images before execution or provisioning.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected for image signatures. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used as image-signing or metadata formats.
REQ-H-IMG-INT-003	Integrity and authenticity verification of guest VM images using protected trust material.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
			where used as image-signing or metadata formats.
REQ-H-RA-INT-003	Cryptographically verifiable attestation evidence bound to the Hypervisor instance, including protection of attestation keys and measurement data against unauthorized access, disclosure or replay.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where selected for attestation signatures. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used as attestation evidence formats.
REQ-H-ADMIN-AUTH-002	Cryptographic key-based administrative authentication and protection of private keys used for administrative authentication.	ACM 5.3 Asymmetric Entity Authentication Schemes; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where used for administrative authentication credentials.
REQ-H-ADMIN-AUTH-003	Strong administrative authentication, with certificate validity, expiration and revocation verification where certificate-based authentication is used.	ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.6.1 Ed25519/Ed448, only where used in the selected certificate/authentication profile.
REQ-H-SERV-AUTH-002	Certificate-based mutual authentication for external services accessing management functions or data.	ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.6.1 Ed25519/Ed448, only where used in the selected certificate/authentication profile.
REQ-H-SERV-AUTH-003	Certificate-based mutual authentication with certificate-path validation, service-identity binding and revocation checking.	ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.6.1 Ed25519/Ed448, only where used in the selected certificate/authentication profile. K.3.2.10.1 IEEE 802.1AR [15], only where secure device/platform identity is used for service identity.
REQ-H-CONF-002	Encryption of sensitive data at rest and protection of data-at-rest encryption keys.	ACM 3.2 Specific Confidentiality Modes: Disk Encryption; ACM 3.1 Confidentiality Modes of Operation: Encryption/Decryption Modes; ACM 8 Key Management	Not applicable unless a CES/VES-specific mechanism is explicitly selected under K.3.2.
REQ-H-CONF-003	Encrypted communication for management interfaces, VM migration traffic and Hypervisor-mediated protected communication channels, with prevention of fallback to unencrypted or weak channels.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-H-LOG-002	Secure export of audit logs to external log-management or SIEM systems using authenticated and encrypted communication channels.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected for log export.
REQ-H-LOG-003	Cryptographic protection, trusted time-stamping or tamper-evident protection enabling verification of the integrity and time or sequence of audit records.	ACM 5.2 Digital Signature; ACM 3.3 Integrity Modes: Message Authentication Codes; ACM 2.3 Hash Functions	K.3.2.9.1 NTS for NTP, only where NTP time is used for security-relevant timestamps. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where logs or claims are protected using those formats.
REQ-H-UPD-001	Authenticity and integrity verification of Hypervisor updates before installation or activation, using signatures, certificates, digests, trusted reference values or equivalent metadata.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected for update signatures. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used as update metadata/signature formats.

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-H-UPD-002	Integrity and authenticity verification of rollback image or package before authorized rollback.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for rollback metadata/signatures.

Y.3 Mapping of Management and Orchestration requirements

Table Y.3-1: Mapping of Management and Orchestration cryptographic requirements

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-M&O-CFG-001	Enforced secure communication protocols as part of the M&O secure default settings.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-M&O-COM-001	Confidentiality and integrity protection of M&O management and orchestration communications.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-M&O-COM-002	Certificate-based mutual authentication for M&O communications with managed hypervisors.	ACM 6.1 TLS; ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.8.1 QUIC, only where QUIC is selected. K.3.2.10.1 IEEE 802.1AR [15], only where secure device/platform identity is used for managed-hypervisor identity.
REQ-M&O-COM-003	Cryptographically protected and mutually authenticated control operations toward verified and authorized managed hypervisors.	ACM 6.1 TLS; ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.8.1 QUIC, only where QUIC is selected. K.3.2.10.1 IEEE 802.1AR [15], only where secure device/platform identity is used.
REQ-M&O-INT-001	Integrity verification of orchestration artefacts against a trusted reference value, such as a cryptographic digest, digital signature or equivalent integrity reference.	ACM 2.3 Hash Functions; ACM 5.2 Digital Signature	K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used as signed artefact/metadata formats.
REQ-M&O-INT-002	Integrity and authenticity verification of orchestration artefacts using digital signatures validated against trusted keys, certificates or equivalent trust anchors.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for signed artefact/metadata formats.
REQ-M&O-INT-003	Integrity and authenticity validation of plugins/extensions/modules before activation, including registration and revocation of trusted signing keys.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for plugin/module signatures or metadata.
REQ-M&O-LOG-002	Secure export of logs to external logging or SIEM systems using encrypted, authenticated protocols.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected for log export.
REQ-M&O-LOG-003	Cryptographic signing and time-stamping of audit logs to verify origin and integrity.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.9.1 NTS for NTP, only where NTP time is used for security-relevant timestamps.

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-M&O-UPD-001	Authenticity and integrity verification of M&O updates before installation, activation, distribution or orchestration.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for update metadata/signatures.
REQ-M&O-UPD-002	Integrity and authenticity verification of rollback images or packages before authorized rollback.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for rollback metadata/signatures.
REQ-M&O-DECOM-002	Confidentiality and integrity protection of transferred orchestration data, management configuration or metadata.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected for protected transfer.

Y.4 Mapping of CRS requirements

Table Y.4-1: Mapping of CRS cryptographic requirements

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-CRS-CP-ISO-003	Cryptographically protected and mutually authenticated communication channels for remote or network-exposed CRS management/control interfaces.	ACM 6.1 TLS; ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-CRS-NP-ISO-003	Cryptographically protected and mutually authenticated management/control traffic over shared or untrusted networks.	ACM 6.1 TLS; ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-CRS-B-INT-002	Integrity and authenticity verification of CRS executable artefacts before initialization or workload execution.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used as signed metadata/protection formats.
REQ-CRS-IMG-INT-001	Integrity verification of container images and associated metadata against trusted reference values, such as manifests or digests.	ACM 2.3 Hash Functions	Not applicable.
REQ-CRS-IMG-INT-002	Integrity and authenticity verification of container images before execution or provisioning, including trusted provenance where applicable.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for image/provenance protection.
REQ-CRS-IMG-INT-003	Integrity and authenticity verification of container images using protected trust material.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for image/provenance protection.

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-CRS-RA-INT-003	Cryptographically verifiable attestation evidence bound to the CRS instance, including protection of attestation keys and measurement data.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where selected for attestation signatures. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used as attestation evidence formats.
REQ-CRS-ADMIN-AUTH-001	Non-reversible, salted password hashing for administrative passwords where password-based authentication is used.	ACM 3.8 Password Protection/Password Hashing Mechanisms	K.3.2.4.1 Argon2id, where selected for password hashing. K.4.2.1.1 scrypt, only for the identified interoperability or compatibility purpose.
REQ-CRS-ADMIN-AUTH-002	Cryptographic key-based administrative authentication and protection of private keys used for administrative authentication.	ACM 5.3 Asymmetric Entity Authentication Schemes; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where used for administrative authentication credentials.
REQ-CRS-ADMIN-AUTH-003	Strong administrative authentication, with certificate validity, expiration and revocation verification where certificates are used.	ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.6.1 Ed25519/Ed448, only where used in the selected certificate/authentication profile.
REQ-CRS-SERV-AUTH-002	Mutual cryptographic authentication for service-to-CRS interactions, with protection of cryptographic materials such as private keys, certificates or tokens.	ACM 5.3 Asymmetric Entity Authentication Schemes; ACM 8 Key Management	K.3.2.7.2 JOSE/JWS/JWE, only where signed or protected tokens are used. K.3.2.6.1 Ed25519/Ed448, only where selected for credentials/signatures.
REQ-CRS-SERV-AUTH-003	Verifiable binding of service identities to cryptographic credentials, including issuance, rotation, revocation and auditability.	ACM 5.3 Asymmetric Entity Authentication Schemes; ACM 8 Key Management	K.3.2.7.2 JOSE/JWS/JWE, only where signed or protected tokens are used. K.3.2.10.1 IEEE 802.1AR [15], only where secure device/platform identity is used.
REQ-CRS-CONF-002	Encryption of sensitive data at rest managed by the CRS and protection of data-at-rest encryption keys.	ACM 3.2 Specific Confidentiality Modes: Disk Encryption; ACM 3.1 Confidentiality Modes of Operation: Encryption/Decryption Modes; ACM 8 Key Management	Not applicable unless a CES/VES-specific mechanism is explicitly selected under K.3.2.
REQ-CRS-CONF-003	Encrypted communication for CRS management/control interfaces, image registries and CRS APIs handling sensitive data or control operations.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-CRS-CONF-004	Secure provisioning of secrets and key material into protected execution environments for confidential containers.	ACM 8 Key Management; ACM 3.6 Key Protection	K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where attestation evidence or claims are used to authorize secret release.
REQ-CRS-LOG-002	Secure export of CRS audit logs using authenticated and encrypted communication channels.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected for log export.

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-CRS-LOG-003	Cryptographic protection, trusted time-stamping or tamper-evident audit-log protection enabling verification of integrity and time or sequence.	ACM 5.2 Digital Signature; ACM 3.3 Integrity Modes: Message Authentication Codes; ACM 2.3 Hash Functions	K.3.2.9.1 NTS for NTP, only where NTP time is used for security-relevant timestamps. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used to protect log evidence/claims.
REQ-CRS-UPD-001	Authenticity and integrity verification of CRS updates before installation or activation.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for update metadata/signatures.
REQ-CRS-UPD-002	Integrity and authenticity verification of rollback image or package before authorized rollback.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for rollback metadata/signatures.

Y.5 Mapping of Container Engine requirements

Table Y.5-1: Mapping of Container Engine cryptographic requirements

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-CE-INTEG-001	Integrity verification of container images and associated metadata before making images available for execution.	ACM 2.3 Hash Functions	Not applicable.
REQ-CE-INTEG-002	Authenticity and integrity verification of container images using digital signatures validated against trusted keys, certificates or equivalent trust anchors.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for image signatures or metadata.
REQ-CE-INTEG-003	Authenticity and integrity verification of container images using protected trust material.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for image signatures or metadata.
REQ-CE-AUTH-002	Cryptographic key-based mutual authentication for communication with the orchestrator or authorized agents, with private-key protection.	ACM 5.3 Asymmetric Entity Authentication Schemes; ACM 8 Key Management	K.3.2.6.1 Ed25519/Ed448, only where selected for authentication credentials.
REQ-CE-AUTH-003	Certificate-based mutual authentication for all management interfaces, with PKI validation and revocation checking.	ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.6.1 Ed25519/Ed448, only where used in the selected certificate/authentication profile.
REQ-CE-CONF-002	Encryption at rest for container images, configuration data and secrets under CE control, and encrypted channels for CE administrative interfaces and communications with external services.	ACM 3.2 Specific Confidentiality Modes: Disk Encryption; ACM 3.1 Confidentiality Modes of Operation: Encryption/Decryption Modes; ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected for encrypted channels.

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-CE-CONF-003	Secure provisioning of secrets and key material into protected execution contexts without exposure outside the protected provisioning path.	ACM 8 Key Management; ACM 3.6 Key Protection	K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where attestation evidence or claims are used to authorize secret release.
REQ-CE-UPD-001	Authenticity and integrity verification of CE updates before installation or activation.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for update metadata/signatures.
REQ-CE-UPD-002	Integrity and authenticity verification of rollback image or package before authorized rollback.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for rollback metadata/signatures.
REQ-CE-DECOM-002	Integrity protection and prevention of unauthorized disclosure or modification during transfer of container image metadata, configuration or management data.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected for protected transfer.

Y.6 Mapping of Container Orchestrator requirements

Table Y.6-1: Mapping of Container Orchestrator cryptographic requirements

Requirement	Cryptographic objective	ACM clause number and title	CES/VES Annex K reference, where applicable
REQ-CO-ISO-003	Cryptographic protection of CO control-plane communications.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-CO-INTEG-002	Integrity verification of CO control-plane component binaries before operation using cryptographic digests, signatures or equivalent integrity metadata.	ACM 2.3 Hash Functions; ACM 5.2 Digital Signature	K.3.2.6.1 Ed25519/Ed448, only where selected for signatures.
REQ-CO-AUTH-002	Cryptographically verifiable authentication mechanisms for service accounts and administrative identities, including mutual authentication for internal control-plane communication.	ACM 5.3 Asymmetric Entity Authentication Schemes; ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected. K.3.2.7.2 JOSE/JWS/JWE, only where signed/protected tokens are used.
REQ-CO-AUTH-003	Certificate-based authentication across control-plane and administrative interfaces with trusted PKI integration.	ACM 5.3 Asymmetric Entity Authentication Schemes	K.3.2.6.1 Ed25519/Ed448, only where used in the selected certificate/authentication profile. K.3.2.10.1 IEEE 802.1AR [15], only where secure device/platform identity is used.
REQ-CO-CONF-001	Protection of control-plane secrets, including authentication tokens, private keys and service credentials, ensuring they are not stored in plaintext and are accessible only through a controlled interface or KMS.	ACM 8 Key Management; ACM 3.6 Key Protection; ACM 3.2 Specific Confidentiality Modes: Disk Encryption	K.3.2.7.2 JOSE/JWS/JWE, only where tokens or claims are protected using JOSE/JWS/JWE.
REQ-CO-CONF-002	Encryption of control-plane traffic and orchestrator-managed communication channels.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected.
REQ-CO-UPD-001	Authenticity and integrity verification of CO updates before installation or activation.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for update metadata/signatures.
REQ-CO-UPD-002	Integrity and authenticity verification of rollback image or package before authorized rollback.	ACM 5.2 Digital Signature; ACM 2.3 Hash Functions	K.3.2.6.1 Ed25519/Ed448, only where selected. K.3.2.7.1 COSE or K.3.2.7.2 JOSE/JWS/JWE, only where used for rollback metadata/signatures.
REQ-CO-DECOM-002	Confidentiality and integrity protection of transferred workload specifications, orchestration configuration, policy objects or metadata.	ACM 6.1 TLS	K.3.2.8.1 QUIC, only where QUIC is selected for protected transfer.

Annex Z (informative): Change history

Date	Version	Information about changes
June 2025	V0.0.1	• Added representative use cases for hypervisors and container runtime systems, classified by risk level. • Defined cumulative security levels (Basic, Elevated, Advanced) to guide requirement selection. • Defined threats • Defined generic security requirements • Defined instantiated security requirements for isolation, integrity, authentication, authorization and confidentiality with assigned security levels across product types (Type 1/2 Hypervisors, CRS). • Added Annex A: Risk assessment methodology • Added Annex B: Mapping between GR requirements and CRA. • Added Annex C: Threat-to-Requirement-to-Control mapping table for traceability. • Added Annex D: Illustrative security controls

August 2025	V0.0.2	• Extend the scope to cover the entire container execution stack, including the Container Runtime System (CRS), the Container Engine (CE), and the Container Orchestrator (CO).
		• Extend the scope to cover the entire Virtualisation execution stack, including the hypervisor and the orchestration and management system.
		• Update scope figures.
		• Replace "security level" with "security category level".
		• Add new requirements specific to the container engine (CE) and container orchestrator (CO).
		• Revise existing CRS requirements to align with the extended scope and updated terminology.
		• Update the risk assessment methodology in Annex A to:
		  o Clarify how threats are factored into risk determination.
		  o Describe how to assess the overall product risk level.
		• Introduce an "Assessment" clause, including an example test case for one of the defined security requirements.
		• Replace the "Assumptions" clause with a new "Operational Environment Objectives" clause.

August 2025	V0.0.3	• Updated scope figures.
		• Revised list of threats, now with specific threats defined per component.
		• New clause on Objectives for the Operational Environment, replacing the previous *Assumptions* clause.
		• New and updated requirements, including:
		  o New requirements for the management and orchestration system.
		  o Updates to isolation and integrity requirements for hypervisors.
		• Addition of three options for structuring the assessment of requirements:
		  o Test case-based approach
		  o Simplified PT2
		  o ETSI TS 103 645 / ETSI EN 303 645 approach

Date	Version	Information about changes
September 2025	V0.0.4	• Addressed new comments received (see Excel sheet v5). • Updated the titles of representative use cases in Clause 5. • Updated requirement categorization and selection approaches in Clause 6: o The label Enhanced has been replaced by Advanced. o A new option for creating a custom security profile has been introduced for cases where the product use case does not fit one of the representative use cases in the present document, or where the manufacturer wishes to conduct a dedicated risk assessment. o A categorization approach: Cumulative, Exclusive, and Single - has been introduced to clarify the risk-based selection of requirements. • Transformed generic requirements into security objectives (Clause 8). • Updated VES and CES requirements following received comments; additional elements have been added, and the new categorization approach has been applied to a subset of them (see Clauses 10 and 11). • Added new requirements on vulnerability handling and SBOM in Clause 12. • Updated the risk assessment methodology: simplification of risk factors and application of the updated risk factors to the assessment of use cases. • Moved implementation notes intended as guidance for manufacturers from the requirements clauses to Annex D.
September 2025	V0.0.5	• Addressed new comments received from ETSI. • Added a time synchronization objective and related requirements for the hypervisor, with implementation notes included in Annex D. • Completed and updated authentication requirements for the hypervisor, covering both administrative and service authentication. • Annex A: CRA essential requirements have been mapped to the existing hypervisor requirements. • Annex C: Now designated as a normative annex. Threats to hypervisor requirements are partially mapped as a sample for presentation; this mapping will be completed once a stable set of requirements with fixed IDs is available. • Reordered clauses: The clause defining labels, categories, and SCLs, along with the approaches for applying requirements, has been moved to appear just before the requirements clause, making the present document easier to follow. • Updated use case descriptions, including illustrative business applications and revised example environments. • Requirement categorization, labelling, and classification approaches have been clarified for manufacturers. The guidance is now more direct and less perspective-based. • The label Standard has been replaced by Elevated.
October 2025	V0.0.6	• Conformance assessment approach described, including evaluation of hypervisor isolation and boot chain integrity requirements. • Incorporated new comments received. • Added definitions of Tenant and Multi-Tenancy. • Slightly updated use cases UC-V3 and UC-C3 to also cover multi-tenancy scenarios within the same organization. • Updated risk assessment methodology applied to UC-V1/2/3, with risk evaluated per threat for hypervisors. • Mapped hypervisor threats to corresponding technical requirements to CRA requirements. • Moved implementation notes to Annex D. • Replaced the ‘Label' term with ‘Class'.
October 2025	V0.0.7	• Update following the ETSI/CEN-CENELEC rapporteurs meeting on the alignment of CRA vertical standards: o Document structure made more compact and logically sequenced. o Assessment approach optimized for consistency. • Addition of a new clause: "Product variants" clarifying the scope of products under the present document, identifying core mandatory components, and distinguishing optional elements. • Update of use cases to remove inconsistencies and confusion related to the terms single-tenant and multi-tenant; consolidation achieved through additional illustrative examples. • All requirements in this version now follow a harmonised structure, with the class-based approach applied consistently across all defined requirements. • Update of requirements leveraging hardware-based technologies to ensure they are described in a more neutral, technology-agnostic, and risk-based manner.

Date	Version	Information about changes
		• CRS requirements are completed. • Addition of a new table clarifying the applicability of hypervisor requirements to Type I, Type II, and Hybrid architectures. • Application of the new assessment approach to both Hypervisor and CRS requirements related to Isolation and Integrity Protection. • Completion of the mapping between CRA essential requirements and the corresponding Hypervisor and CRS requirements. Completion of the mapping between identified Threats and corresponding Requirements for both Hypervisor and CRS components.
November 2025	V0.0.8	• Addressed most of the comments received during the pre-open consultation. • Addressed most of the comments received from the ETSI team working on the present document. • Updated the use cases section, including the addition of edge deployment scenarios. • Clarified the assessment approach and added requirements for the assessment report and the evidence package. • Added applicability notes to several security requirements where relevant. • Added a new clause, "Roles and Responsibilities," clarifying the responsibilities of the manufacturer and the product integrator. • Added additional assessment cases for confidentiality and availability requirements for Hyper and CRS components. • Updated the applicability table for hypervisor requirements, replacing "partially applicable" with "conditionally applicable." • Added references to prEN 40000-1-1, prEN 40000-1-2, and prEN 40000-1-3. • Referenced prEN 40000-1-3 for SBOM and vulnerability handling requirements. • Updated several requirements following received comments, including changes in classification for some of them.
November 2025	V0.0.9	• Added a new hardware-related threat and performed the corresponding risk assessment. • Updated the applicability table for requirements on Type I, Type II and Hybrid hypervisors to clarify that a requirement may be implemented directly by the hypervisor or delegated to other OE components. • Updated the isolation, integrity and confidentiality requirements: Advanced class may leverage software- or hardware-based mechanisms; hardware-assisted requirements are now optional ("may"). • Completed the assessment descriptions for all Hypervisor and CRS requirements. • Updated the risk assessment for the use cases and refined the rationales to better reflect the context of the risk factors and use cases. • Performed an end-to-end alignment in preparation for the HAS review.
December 2025	V0.0.10	• Revised based on ETSI editing support comments to improve compliance with ETSI drafting requirements.
March 2026	V0.0.11	• Update based on OC1/2/3 comments • Update based on the HAS comments received on Clause 4 (except comments #7 and #16)
March 2026	V0.0.12	• Update based on the HAS comments received on Clause 4/5
April 2026	V0.0.13	• Update based on the HAS comments received on Clause 4/5 • Update based on stakeholders' comments
April 2026	V0.0.14	• Update based on the HAS comments received on Clause 5 • Update based on stakeholders' comments (Google, XCP-ng OSI) • Addition of mapping between requirements and crypto operations from ACM • Alignment with the cross vertical skeleton
April 2026	V0.0.15	• Clauses 5 and 6 updated based on the received HAS comments • Document updated based on stakeholders' comments, including XCP-ng / OSI feedback • Terminology updated: SCL replaced by Security Profile (SP) • Technology-specific terms (TEE / TPM / HSM) replaced by generic wording • Threat-to-requirement mapping tables moved to Annex B • Applicability tables for UCs / SPs added under each requirements subclause • Document structure aligned with the cross-vertical skeleton • Annex R (RDPS) added as a placeholder, pending integration of a stable and endorsed version

Date	Version	Information about changes
May 2026	V0.0.16	• Clauses 5, 6 and Annex A updated based on the received HAS comments • Assessment cases completed for the M&O, CE, and CO components • Assessment cases completed for the assurance requirement set • Existing assessment cases updated to reflect the revised requirements • Decommissioning and secure deletion requirements added in relation to CRA Annex I, Part I, point (2)(m) • Threat-to-requirement mappings completed for the M&O, CE, and CO components • Annex A revised in line with the updates made to clause 5
May 2026	V0.0.17	• Open Consultation comments addressed. • Part of the comments received on version 16 addressed. • Annex X added, covering vertical-specific cryptographic requirements. • Annex Y added, providing the mapping between the requirements and the Agreed Cryptographic Mechanisms (ACM).
June 2026	V1.0.0	• Final draft version submitted for endorsement.

History

Version	Date	Status	
V1.0.1	June 2026	SRdAP process	EV 20260916: 2026-06-25 to 2026-09-16