ETSI TS 133 117 V17.1.0 (2022-09)

TECHNICAL SPECIFICATION

Universal Mobile Telecommunications System (UMTS); LTE; 5G; Catalogue of general security assurance requirements (3GPP TS 33.117 version 17.1.0 Release 17)

Reference

RTS/TSGS-0333117vh10

Keywords

5G,LTE,SECURITY,UMTS

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:
http://www.etsi.org/standards-search

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 at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

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.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
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

Intellectual Property Rights

Essential patents

Trademarks

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

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.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document contains objectives, requirements and test cases that are deemed applicable, possibly after adaptation, to several network product classes.

Several network product classes share very similar if not identical security requirements for some aspects. Therefore, these are collected in this "catalogue" document applicable to many network product classes. In addition to this catalogue, requirements specific to different network product classes will be captured in separate documents.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TR 41.001: "GSM Specification set".

[3] IETF RFC 3871: "Operational Security Requirements for Large Internet Service Provider (ISP) IP Network Infrastructure".

[4] 3GPP TR 33.926: "Security Assurance Specification (SCAS) threats and critical assets in 3GPP network product classes".

[5] CVE-1999-0511, http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-1999-0511

[6] "Practical recommendations for securing Internet-connected Windows NT Systems", https://support2.microsoft.com/default.aspx?scid=kb;%5BLN%5D;164882.

[7] X-Force Vulnerability Report, http://www.iss.net/security_center/static/193.php

[8] IETF RFC 2644: "Changing the Default for Directed Broadcasts in Routers."

[9] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[10] 3GPP TS 33.501 v15: "Security architecture and procedures for 5G system".

[11] IETF RFC 7540: "Hypertext Transfer Protocol Version 2 (HTTP/2)".

[12] IETF RFC 6749: "OAuth2.0 Authorization Framework".

[13] 3GPP TS 29.501: "Principles and Guidelines for Services Definition".

[14] 3GPP TS 33.501: "Security architecture and procedures for 5G system" (Release 16).

[15] 3GPP TS 33.2:10: "Network Domain Security (NDS); IP network layer security".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

Machine Accounts: These will be used for authentication and authorization from system to system or between applications on a system and cannot be assigned to a single person or a group of persons.

Personal data: any information relating to an identified or identifiable natural person ('data subject').

Identifiable person: one who can be identified, directly or indirectly, in particular by reference to an identification number, name or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity.

NOTE: personal data can be gathered from user data and traffic data.

Sensitive data: data that may be used for authentication or may help to identify the user, such as user names, passwords, PINs, cryptographic keys, IMSIs, IMEIs, MSISDNs, or IP addresses of the UE, as well as files of a system that are needed for the functionality such as firmware images, patches, drivers or kernel modules.

System group account: a predefined system account in the network product, usually with special privileges, which has a predefined user id and hence cannot be tied to a single user (individual) in a normal operating environment.

EXAMPLE: the 'root' account.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

API	Application Programming Interface
CIS	Center for Internet Security
JSON	Java Script Object Notation
NF	Network Function
NRF	Network Repository Function
SBA	Service Based Architecture
SBI	Service Based Interfaces
SEPP	Security Edge Protection Proxy
URI	Uniform Resource Identifier
WAS	Web Application Security

4 Catalogue of security requirements and related test cases

4.1 Introduction

4.1.1 Pre-requisites for testing

The SCAS tests, as described in the present specification, are to be applied to a network product whose software and hardware has been brought into use so that the network product can provide the intended functionality, either in a real network environment or in a simulated environment. This implies that, before any testing is performed, the hardware and software has been installed correctly, the network product is powered on, and communication has been established over all standardized interfaces and OAM interfaces related with the network product's functionality, as described in the vendor's documentation.

Communication over external non standardized Interfaces that may exist and are marked as optional, according to the vendor's documentation, shall also be established during testing unless they are explicitly marked as "not recommended" in the vendor's documentation.

For each of the enabled external communication interfaces there may be various optional capabilities. During testing, all such capabilities shall be enabled unless they are explicitly marked as "not recommended" in the vendor's documentation.

In some cases a testcase might require configuration changes as part of the execution steps or pre-conditions. After such test is executed and prior to any further test execution it needs to be ensured that the state of the ToE is restored back in the original state.

SCAS testing is not about security in operations and deployments. So, in particular, SCAS testing is independent of any operator guidelines or considerations on specific deployment scenarios.

4.1.2 Use of tools in testing

The following text shall apply to all test cases described in the present document:

The present document takes into account that the landscape of testing tools evolves more rapidly than SCAS specifications. It is therefore allowed that, for each requirement, the actual test carried out may deviate from the stepwise description of the test case in the present document if the following conditions are fulfilled:

(1) The test is carried out by preferably using Commercial-of-the-Shelf (COTS) and Free-Open-Source-Software (FOSS) tools that are available for other testers that may want to repeat the test. In case a tool not in any of these two categories is used then evidence of the quality assurance of the tool needs to be provided. This applies only to tools used to perform the actual test and not supportive tools needed for setting up the testing environment like for example traffic generators/ simulators.

In cases where a test lab is not able to obtain the necessary tools to perform the test, vendor proprietary test tools may be used by the test lab as long the test tool is controlled under a suitable quality management system (QMS). The test lab ensures that this QMS is in place in order to avail of a vendor’s test tool.

Additionally in cases where the accredited test lab does not have the necessary test environment to perform a test, it shall be possible for the accredited test lab personnel to perform the test in a vendor's test lab. In such cases the accredited lab should record details of test environment, test set-up used and how the test was performed.

(2) The tester provides evidence, e.g. by referring to the documentation of the tool, that the tool is suitable to verify the requirement, and the scope of testing is equal or larger to the one of the test case described in the present document. The evidence needs to be sufficiently detailed for experts in the field of testing, not for the general public.

(3) The tester provides evidence that the tool has been actually used for testing the network product (e.g. by providing a trace).

4.1.3 Documentation Requirements

When a test case makes an assumption on the availability of certain items in the product documentation then this assumption is to be considered part of the requirement even if the requirements text does not mention the documentation.

4.2 Security functional requirements and related test cases

4.2.1 Introduction

The present clause describes the security functional requirements and the corresponding test cases, independent of a specific network product class. In particular the proposed security requirements are classified in two groups:

- Security functional requirements deriving from 3GPP specifications and detailed in clause 4.2.2
- General security functional requirements which include requirements not already addressed in the 3GPP specifications but whose support is also important to ensure a network product conforms to a common security baseline detailed in clause 4.2.3.

4.2.2 Security functional requirements deriving from 3GPP specifications and related test cases

4.2.2.1 Security functional requirements deriving from 3GPP specifications – general approach

The present clause describes the general approach taken towards security functional requirements deriving from 3GPP specifications and the corresponding test cases, independent of a specific network product class.

It is assumed for the purpose of the present SCAS that a network product conforms to all mandatory security-related provisions in 3GPP specifications pertaining to it, in particular:

- all 3GPP specifications of the 33-series (security specifications) that are pertinent to the network product class;
- other 3GPP specifications that make reference to security specifications or are referred to from one of them.

3GPP has decided to develop test specifications for the UE in the TSs of the 34-series under the responsibility of Working Group RAN5. 3GPP saw, however, no need to develop test specifications for network elements. For network elements, 3GPP rather trusts that tests are run under the responsibility of the vendors.

Security procedures pertaining to a network product are typically embedded in non-security procedures and are hence assumed to be tested together with them.

It is the purpose of the present SCAS to identify security requirements from the EPS and 5G security architecture that require special attention in testing as they may:

(a) lead to vulnerabilities when not satisfied;

(b) not be captured through ordinary testing activity for non-security procedures;

(c) address security-relevant failure cases and exceptions or 'negative' requirements of the kind: "The network product shall not…"

It is not an intention of the present document to provide an exhaustive set of test cases that would be sufficient to demonstrate conformance of all security procedures with the above-mentioned specifications.

4.2.2.*2* Security functional requirements derived from 3GPP specifications – general SBA/SBI aspects

4.2.2.*2*.1 Introduction

The purpose of the sub-clauses in 4.2.2.2 is to identify and describe the general baseline requirements from SBA security architecture and the corresponding test cases. The general baseline requirements are applicable to all Network Function (NF) within the 5G Core (5GC) utilizing Service-Based Interfaces (SBI), independent of a specific network product class.

4.2.2.*2*.2 Protection at the transport layer

Requirement Name: Protection at the transport layer

Requirement Reference: TS 33.501 [10], clause 5.9.2.1, clause 13.1, clause 13.3.2

Requirement Description:

"NF Service Request and Response procedure shall support mutual authentication between NF consumer and NF producer" as specified in TS 33.501 [10], clause 5.9.2.1;

"All network functions shall support TLS. Network functions shall support both server-side and client-side certificates.

The TLS profile shall follow the profile given in Annex E of TS 33.310 [9] with the restriction that it shall be compliant with the profile given by HTTP/2 as defined in RFC 7540 [11]. "

as specified in TS 33.501 [10], clause 13.1.

"Authentication between network functions within one PLMN shall use one of the following methods:

- If the PLMN uses protection at the transport layer as described in clause 13.1, authentication provided by the transport layer protection solution shall be used for authentication between NFs."

as specified in TS 33.501 [10], clause 13.3.2.

Threat References: TR 33.926 [4], clause 5.3.6.3, Weak cryptographic algorithms

Test case:

Test Name: TC_PROTECT_TRANSPORT_LAYER

Purpose:

Verify that TLS protocol for NF mutual authentication and NF transport layer protection is implemented in the network products based on the profile required.

Procedure and execution steps:

Pre-Conditions:

Network product documentation containing information about supported TLS protocol and certificates is provided by the vendor.

A peer implementing the TLS protocol configured by the vendor shall be available.

The tester shall base the tests on the profile defined by 3GPP in Annex E of TS 33.310 [9] with the restriction that it shall be compliant with the profile given by HTTP/2 as defined in RFC 7540 [11].

Execution Steps

1. The tester shall check that compliance with the TLS profile can be inferred from detailed provisions in the network product documentation.

2. The tester shall establish a secure connection between the network product under test and the peer and verify that all TLS protocol versions and combinations of cryptographic algorithms that are mandated by the TLS profile are supported by the network product under test.

3. The tester shall try to establish a secure connection between the network product under test and the peer and verify that this is not possible when the peer only offers a feature, including protocol version and combination of cryptographic algorithms, that is forbidden by the TLS profile.

Expected Results:

- The network product under test and the peer establish TLS if the TLS profiles used by the peer are compliant with the profile requirements in TS 33.310 [9] Annex E and RFC 7540 [11].
- The network product under test and the peer fail to establish TLS if the TLS profiles used by the peer are forbidden in TS 33.310 [9] Annex E or RFC 7540 [11].

Expected format of evidence:

Provide evidence of the check of the product documentation in plain text. Save the logs and the communication flow in a .pcap file.

4.2.2.2.3 Authorization of NF service access

4.2.2.2.3.1 Authorization token verification failure handling wthin one PLMN

Requirement Name: Authorization token verification failure handling wthin one PLMN

Requirement Reference: TS 33.501 [14], clause 13.4.1.1

Requirement Description:

"13.4.1.1 Service access authorization within the PLMN

2. The NF Service producer shall verify the access token as follows:

    - The NF Service producer ensures the integrity of the access token by verifying the signature using NRF's public key or checking the MAC value using the shared secret. If integrity check is successful, the NF Service producer shall verify the claims in the access token as follows:

NOTE: Void.

- It checks that the audience claim in the access token matches its own identity or the type of NF service producer. If a list of NSSAIs or list of NSI IDs is present, the NF service producer shall check that it serves the corresponding slice(s).
- If an NF Set ID present, the NF Service Producer shall check the NF Set ID in the claim matches its own NF Set ID.
- If the access token contains "additional scope" information (i.e. allowed resources and allowed actions (service operations) on the resources), it checks that the additional scope matches the requested service operation.
- If scope is present, it checks that the scope matches the requested service operation.
- It checks that the access token has not expired by verifying the expiration time in the access token against the current data/time.

3. If the verification is successful, the NF Service producer shall execute the requested service and responds back to the NF Service consumer. Otherwise it shall reply based on Oauth 2.0 error response defined in RFC 6749 [43]. The NF service consumer may store the received token(s). Stored tokens may be re-used for accessing service(s) from producer NF type listed in claims (scope, audience) during their validity time.

Threat References: TR 33.926 [4], clause 6.3.3.1, Incorrect Verification of Access Tokens

Test Case:

Test Name: TC_AUTHORIZATION_TOKEN_VERIFICATION_FAILURE_ONE_PLMN

Purpose:

Verify that the NF service producer does not grant service access if the verification of authorization token from a NF service consumer in the same PLMN fails.

Procedure and execution steps:

Pre-Conditions:

- Test environment with a NF service consumer.
- The NF service consumer may be simulated.
- The network product under test has already mutually authenticated with the NF service consumer.
- The tester shall have access to the interface between the NF service consumer and the network product under test.
- The tester has the NRF’s private key or the shared key.
- The network product under test is preconfigured with the NRF’s public key or the shared key.

Execution Steps

The network product under test receives the access token sent from the NF service consumer, verifies the access token based on Oauth 2.0.
Test Cases 1~4 are tests on failure handling by the network product under test when the mandatory claims in access token failed verification.

Test Case 1: Verification failure of the access token integrity

1) The tester computes an access token correctly, except that the signature or the MAC is incorrect, e.g., the signature or the MAC is randomly selected, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test.
2) The integrity verification of the access token by the network product under test fails.

Test Case 2: Incorrect audience claim in the access token

1) The tester computes an access token correctly, except that the audience claim is incorrect, i.e., the audience claim in the access token does not match the identity or the type of the network product under test, and then includes the access token in the NF Service Request sent from NF service consumer to the network product under test.
2) The network product under test verifies that the integrity of the access token is valid. However, the audience claim in the access token does not match its identity or type.

Test Case 3: Incorrect scope claim in the access token

1) The tester computes an access token correctly, except that the scope is incorrect, i.e., the scope does not match the requested service operation, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test.
2) The network product under test verifies that the integrity of the access token and the audience claim are valid. However, the scope does not match the requested service operation.

Test Case 4: Expired access token

1) The tester computes an access token correctly, except that the expiration time has expired against the current data/time, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test.
2) The network product under test verifies that the integrity of the access token, the audience and scope claims are all valid. However, the expiration time in the access token has expired against the current data/time.

Test Cases 5~8 are tests on failure handling by the network product under test when the optional claims in access token failed verification.

NOTE: The test cases below only apply to the NFs which support identifying and understanding the optioanl claims in the received access token.

Test Case 5: Incorrect list of S-NSSAIs in the access token

1) The tester computes an access token correctly, except that the list of S-NSSAIs is incorrect, i.e., the network product under test does not serve the slices indicated in the list of S-NSSAIs, and then includes the access token in the NF Service Request sent from NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience, scope and expiration time claims are all valid. Then it further checks the list of S--NSSAIs included in the access token.

Test Case 6: Incorrect list of NSIs in the access token

1) The tester computes an access token correctly, except that the list of NSIs is incorrect, i.e., the network product under test does not serve the slices indicated in the list of NSIs, and then includes the access token in the NF Service Request sent from NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience, scope and expiration time claims are all valid. Then it further checks the list of NSIs included in the access token.

Test Case 7: Incorrect NF Set ID in the access token

1) The tester computes an access token correctly, except that the NF Set ID is incorrect, i.e. the NF Set ID in the claim does not match the NF Set ID of the network product under test, and then includes the access token in the NF Service Request sent from NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience, scope and expiration time claims are all valid. Then it further checks the NF Set ID included in the access token.

Test Case 8: Incorrect additional scope in the access token

1) The tester computes an access token correctly, except that the additional scope information is incorrect, i.e. the allowed resources and allowed actions on the resources do not match the requested service operations, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test.

2) The network product under test verifies that the integrity of the access token, the audience, scope and expiration time claims are all valid. Then it further checks the additional scope included in the access token.

Expected Results:

For test cases 1~4 on verification failure of mandatory claims in the access token, the network product under test rejects the NF service consumer’s service request based on Oauth 2.0 error response defined in RFC 6749 [12].

For test cases 5~8 on verification failure of optional claims in the access token, if the network product under test understands these optional claims (list of S-NSSAIs, list of NSIs, NF Set ID, additional scope), it rejects the NF service consumer’s service request based on Oauth 2.0 error response defined in RFC 6749 [12].

Expected format of evidence:

Evidence suitable for the interface, e.g., Screenshot containing the operational results.

4.2.2.2.3.2 Authorization token verification failure handling in different PLMNs

Requirement Name: Authorization token verification failure handling in different PLMNs

Requirement Reference: TS 33.501 [10], clause 13.4.1.2

Requirement Description:

"The NF service producer shall check that the home PLMN ID of audience claim in the access token matches its own PLMN identity."

Threat References: TR 33.926 [4], clause 6.3.3.1, Incorrect Verification of Access Tokens

NOTE: The test case below only applies to the NFs which support identifying and understanding the producerPlmnId claim.

Test Case:

Test Name: TC_AUTHORIZATION_TOKEN_VERIFICATION_FAILURE_DIFF_PLMN

Purpose:

Verify that the NF service producer does not grant service access if the verification of authorization token from a NF service consumer in a different PLMN fails.

Procedure and execution steps:

Pre-Conditions:

- Test environment with a NF service consumer and two SEPPs (one cSEPP, one pSEPP).
- The NF service consumer and SEPPs may be simulated.
- The network product under test has already mutually authenticated with the NF service consumer in a different PLMN via the SEPPs.
- The tester has the NRF's private key or the shared key.
- The network product under test is preconfigured with the NRF's public key or the shared key.
- The tester shall have access to the interfaces of the NF service consumer and the network product under test.

Execution Steps

The network product under test receives the access token sent from the NF service consumer, verifies the access token in accordance with the execution steps in 4.2.2.1.2.1, with the following additional test cases:

Test Case 1: incorrect PLMN ID of the NF service producer in the access token

1) The test computes an access token correctly, except that the PLMN ID in the producerPlmnId claim of the access token is empty or different from the home PLMN ID of the network product under test, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test through the SEPPs.
2) The network product under test receives the access token sent from the NF service consumer through the SEPPs, verifies that the PLMN ID in the producerPlmnId claim of the access token is different from its own home PLMN identity**.**

Test Case 2: absent PLMN ID of the NF service producer in the access token

1) The test computes an access token correctly, except that no producerPlmnId claim is included in the access token, and then includes the access token in the NF Service Request sent from the NF service consumer to the network product under test through the SEPPs.
2) The network product under test receives the access token sent from the NF service consumer through the SEPPs, verifies that the access token is not a token to be used by the NF service consumer in a different PLMN, based on the absence of PLMN ID of the NF service producer in the access token.

Expected Results:

For both test cases 1 and 2, the network product under test rejects the NF service consumer's service request based on Oauth 2.0 error response defined in RFC 6749 [12].

Expected format of evidence:

Evidence suitable for the interface, e.g., Screenshot containing the operational results.

4.2.2.2.4 Authentication for Indirect Communication

4.2.2.2.4.1 Correct handling of client credentials assertion validation failure

Requirement Name: Correct handling of client credentials assertion validation failure

Requirement Reference: TS 33.501 [14], clause 13.3.8.3

Requirement Description:

"The verification of the Client credentials assertion shall be performed by the receiving node, i.e., NRF or NF Service Producer in the following way:

- It validates the signature of the JWS as described in RFC 7515 [45].
- If validates the timestamp (iat) and/or the expiration time (exp) as specified in RFC 7519 [44].

  If the receiving node is the NRF, the NRF validates the timestamp (iat) and the expiration time (exp).

  If the receiving node is the NF Service Producer, the NF service Producer validates the expiration time and it may validate the timestamp.

- It checks that the audience claim in the the client credentials assertion matches its own type.

It verifies that the NF instance ID in the client credentials assertion matches the NF instance ID in the public key certificate used for signing the assertion".

Threat References: TR 33.926 [4], clause 6.3.x.1, Incorrect validation of client credentials assertion

Note: The following test case only applies if the NF under test implements verification of client credentials assertions.

Test Case:

Test Name: TC_CLIENT_CREDENTIALS_ASSERTION_VALIDATION

Purpose:

Verify that the NF under test correctly handles client credentials assertion validation failure.

Editor's Note: This test case applies for Rel-16 NFs. The formulation for indicating the applicable release may need to be updated.

Procedure and execution steps:

Pre-Conditions:

- Test environment with a consumer NF and a SCP, which may be simulated. (Potentially simulated) consumer NF and (potentially simulated) SCP can be combined for the testing purpose.
- The NF under test is preconfigured with the certificate of the consumer NF.
- The NF under test is configured to require assertions for NF consumer authentication for at least one of its services.
- The tester has the private key of the consumer NF.
- The tester has access to the interface between the consumer NF and the NF under test.

Execution Steps

Test Case 1: Failed verification of the client credentials assertion integrity

1) The tester computes a client credentials assertion correctly, except that the signature is incorrect, and then includes the client credentials assertion in the service request sent from the consumer NF to the NF under test via the SCP.

2) The integrity verification of the client credentials assertion by the NF under test fails.

Test Case 2: Incorrect audience claim in the client credentials assertion

1) The tester computes a client credentials assertion correctly, except that the audience claim is incorrect, i.e., the audience claim in the client credentials assertion does not match the type of the NF under test, and then includes the signed client credentials assertion in the service request sent from the consumer NF to the NF under test via the SCP.

2) The NF under test verifies that the audience claim in the client credentials assertion does not match its type.

Test Case 3: Expired client credentials assertion

1) The tester computes an access token correctly, except that the expiration time (exp) has expired against the current time, and then includes the signed client credentials assertion in the service request sent from the consumer NF to the NF under test via the SCP.

2) The NF under test verifies that the expiration time in the client credentials assertion has expired against the current time.

Expected Results:

For test cases 1~3, the NF under test rejects the consumer NF's service request and sends back an error message.

Editor's Note: the result needs to be aligned with the relevant error handling description to be added in TS 29.500.

Expected format of evidence:

Evidence suitable for the interface, e.g. screenshot containing the operational results.

4.2.3 Technical baseline

4.2.3.1 Introduction

The technical baseline is a generic set of security requirements to be fulfilled by all network products.

In particular these requirements counter the security threats and objectives identified in the TR 33.926 [4] and they basically aim to guarantee the network product confidentiality, integrity and availability.

4.2.3.2 Protecting data and information

4.2.3.2.1 Protecting data and information – general

Adequate security measures for protecting sensitive data shall be implemented as defined in the present document. Further measures (that are beyond the scope of the present document) may be required by local regulation depending on the classification of the data and other factors such as type of network used during transmission, storage location for data, etc.

4.2.3.2.2 Protecting data and information – Confidential System Internal Data

Requirement Name: Unauthorized Viewing

Requirement Description: When the system is not under maintenance, there shall be no system function that reveals confidential system internal data in the clear to users and administrators. Such functions could be, for example, local or remote OAM CLI or GUI, logging messages, alarms, configuration file exports etc. Confidential system internal data contains authentication data (i.e. PINs, cryptographic keys, passwords, cookies) as well as system internal data that is not required for systems administration and could be of advantage to attackers (i.e. stack traces in error messages).

Security Objective references: tba.

Test case:

Test Name: TC_CONFIDENTIAL_SYSTEM_INTERNAL_DATA

Purpose:

Verify that no system function reveals sensitive data in the clear

Procedure and execution steps:

Pre-Condition:

The vendor shall provide documentation describing how confidential system internal information that could possibly be revealed in clear-text is handled by system functions.

A list of all system functions in the network product, information on how to enable and execute them should be provided as a part of the vendor's documentation. A system function is every function implemented in the network product needed by the services/functionalities provided by the network product itself.

Execution Steps

Execute the following steps:

1. Review the documentation provided by the vendor describing how confidential system internal information is handled by system functions.

2. The tester checks whether any system functions as described in the product documentation (e.g. local or remote OAM CLI or GUI, logging messages, alarms, error messages, configuration file exports, stack traces) reveal any confidential system internal data in the clear (for example, passphrases).

Expected Results:

There should be no confidential system internal data revealed in the clear by any system function.

Expected format of evidence:

Evidence suitable for the interface, e.g. screenshot containing the operational results.

4.2.3.2.3 Protecting data and information in storage

Requirement Name: Protecting data and information in storage

Requirement Description:

For sensitive data in (persistent or temporary) storage read access rights shall be restricted. Files of a system that are needed for the functionality shall be protected against manipulation.

In addition, the following rules apply for:

- Systems that need access to identification and authentication data in the clear, e.g. in order to perform an authentication. Such systems shall not store this data in the clear, but scramble or encrypt it by implementation-specific means.

- Systems that do not need access to sensitive data (e.g. user passwords) in the clear. Such systems shall hash this sensitive data .

- Stored files on the network product: examples for protection against manipulation are the use of checksum or cryptographic methods.

Security Objective references: tba

Test case:

Test Name: TC_PSW_STOR_SUPPORT

Purpose:

Verify that Password storage use one-way hash algorithm.

Procedure and execution steps:

Pre-Conditions:

- The tester can access the storage of own user account password.
- The tester has privileges to change the password.
- The original password is P1.

Execution Steps

1. The tester accesses the storage where the result of P1 is, and the corresponding hash value is recorded as A
2. The tester changes the password with P2, then the tester records the storage hash value of the new password as B
3. The tester repeats the step 2 to get other records.
4. The tester verifies whether all the records comply with the characteristic of one-way hash result.

Expected Results:

All records comply with the characteristic of one-way hash result.

Expected format of evidence:

Evidence suitable for the interface, e.g. screenshot containing the operation results.

4.2.3.2.4 Protecting data and information in transfer

Requirement Name: tba

Requirement Description:

- Usage of cryptographically protected network protocols is required.
- The transmission of data with a need of protection shall use industry standard network protocols with sufficient security measures and industry accepted algorithms. In particular, a protocol version without known vulnerabilities or a secure alternative shall be used.

Security Objective references: tba

Test case:

Test Name: TC_PROTECT_DATA_INFO_TRANSFER_1

Purpose:

Verify the mechanisms implemented to protect data and information in transfer to and from the Network Product's OAM interface.

NOTE: The test is limited to the OAM interface although the requirement does not have this limitation because the protection of standardised interfaces will be covered by regular interoperability testing and the proprietary use of HTTPS is covered in clause 4.2.5.1.

Procedure and execution steps:

Pre-Conditions:

Network product documentation containing information about supported OAM protocols is provided by the vendor,

A peer implementing the security protocol configured by the vendor (e.g SSH client supporting SSHv2 or HTTPS client) shall be available.

Network product documentation stating which security protocols for protection of data in transit are implemented and which profiles in TS 33.310 [9] and TS 33.210 [15] are applicable is provided by the vendor

For TLS/DTLS, the tester shall base the tests on the profile defined by 3GPP in TS 33.310 [9] and TS 33.210 [15]. For IKE and IPsec, the tester shall base the tests on the profile defined by 3GPP in TS 33.210 [15]. For protocols, for which 3GPP did not define a security profile, e.g. SSH, the tester shall base the tests on a widely recognised and publicly available security profile.

Execution Steps

1. The tester shall check that compliance with the selected security profile can be inferred from detailed provisions in the product documentation.

2. The tester shall establish a secure connection between the network product and the peer and verify that all protocol versions and combinations of cryptographic algorithms that are mandated by the security profile are supported by the network product.

3. The tester shall try to establish a secure connection between the network product and the peer and verify that this is not possible when the peer only offers a feature, including protocol version and combination of cryptographic algorithms, that is forbidden by the security profile.

Expected Results:

The traffic is properly protected, and insecure options are not accepted by the Network Product.

Expected format of evidence:

Provide evidence of the check of the product documentation in plain text. Save the logs and the communication flow in a .pcap file.

4.2.3.2.5 Logging access to personal data

Requirement Name: Logging access to personal data

Requirement Description:

In some cases access to personal data in clear text might be required. If such access is required, access to this data shall be logged, and the log shall contain who accessed what data without revealing personal data in clear text. When for practical purposes such logging is not available, a coarser grain logging is allowed.

In some cases, the personal data stored in the log files may allow the direct identification of a subscriber. In such cases, the revealed personal information may not expose the subscriber to any kind of privacy violation.

Test case:

Test Name: TC_LOGGING_ACCESS_TO_PERSONAL_DATA

Purpose:

Verify that in cases where a network product presents personal data in clear text, access attempts to such data are logged and the log information includes the user identity that has accessed the data. The test case also verifies that the personal data itself is not included in clear text in the log.

Procedure and execution steps:

Pre-Conditions:

A document which provides a description of where personal data in clear text is accessible on the network product, how it can be accessed, and details of where such access attempts are logged and how to view these logs.

Execution Steps

- The tester verifies, for cases where personal data is accessible in clear text, that attempts to access it are recorded in a log, that the log includes the identity of the user that has attempted to access the data, and that the log does not include the actual personal data in clear-text.
- The tester repeats the check for each case where personal data is accessible.

Expected Results:

All access attempts to personal data (in clear text) are recorded in the described logs, with the user identity included and no personal data visible in the log.

Expected format of evidence:

Sample copies of the log files.

4.2.3.3 Protecting availability and integrity

4.2.3.3.1 System handling during overload situations

Requirement Name: System handling during overload situations

- *Requirement Description:*

The system shall provide security measures to deal with overload situations which may occur as a result of a denial of service attack or during periods of increased traffic, or reach the congestion threshold. In particular, partial or complete impairment of system availability shall be avoided. Potential protective measures include:

- Restricting available RAM per application.
- Restricting maximum sessions for a Web application.
- Defining the maximum size of a dataset.
- Restricting CPU resources per process.
- Prioritizing processes.
- Overload control method, e.g. limiting amount or size of transactions of a user or from an IP address in a specific time range.

- *Security Objective references*: tba.

Test case: Refer to test case in clause 4.2.3.3.3.

4.2.3.3.2 Boot from intended memory devices only

Requirement name: Boot from intended memory devices only

Requirement reference: to be done later

Requirement Description:

The network product can boot only from the memory devices intended for this purpose.

Test case:

Test Name: TC_BOOT_INT_MEM_1

Purpose:

Verify that the network product can only boot from memory devices intended for this purpose (e.g. not from external memory like USB key).

Procedure and execution steps:

Pre-Conditions:

A document which contains information regarding the firmware access mechanism supported by the product and about the memory devices from which the network product can boot.

Execution Steps

1. The tester verifies that the network product is configured to boot from memory devices declared in the network product document only.
2. The tester verifies that there is no possibility to access and modify the firmware of the network product without successful authentication.

Expected Results:

The network product cannot boot from a memory device that is not configured in its firmware, and access to the firmware is only possible with the correct authentication.

Expected format of evidence: NA

4.2.3.3.3 System handling during excessive overload situations

Requirement Name: System handling during excessive overload situations

Requirement Description: The system shall act in a predictable way if an overload situation cannot be prevented. A system shall be built in this way that it can react on an overload situation in a controlled way. However it is possible that a situation happens where the security measures are no longer sufficient.

In such case it shall be ensured that the system cannot reach an undefined and thus potentially insecure state. In an extreme case this means that a controlled system shutdown is preferable to uncontrolled failure of the security functions and thus loss of system protection.

The vendor shall provide a technical description of the network product's Over Load Control mechanisms (especially whether these mechanisms rely on cooperation of other network elements e.g. eNode B) and the accompanying test case for this requirement will check that the description provides sufficient detail in order for an evaluator to understand how the mechanism is designed.

Security Objective references: tba.

Test case:

Test Name: TC_SYSTEM_HANDLING_OF_OVERLOAD_SITUATIONS

NOTE: This test case covers requirements 4.2.3.3.1 and this requirement4.2.3.3.3.

Purpose:

Verify that the network product:

- has a detailed technical description of the overload control mechanisms used to deal with overload scenarios;
- has test results verifying the operation of the overload control mechanisms.

Procedure and execution steps:

Pre-Conditions:

- A document which provide a detailed technical description of the overload control mechanisms.
- Test results from a test execution phase of overload control mechanism testing.

Execution Steps

- The tester verifies that there is:
    - A technical description providing a high-level overview of the overload control design:
        - An overview of the types of overload scenarios that the network product overload control mechanisms are expected to handle.
        - An overview of the overload control thresholds that the network product uses to trigger overload control mechanisms.
        - Description of the types of attacks that may cause an overload to the network product and how these are handled.
        - A description of how the network product discards or handles input during various overload situations including excessive overloads. i.e. where the overload is significantly greater than the thresholds where overload detection is triggered.
        - A description of how the network product security functions operate and perform during overload.
        - A description of how the network product shuts down or performs or takes other abatement or corrective actions during excessive overload conditions.
    - The tester verifies that the test results:
        - Contain details of the overload conditions used in the test execution that are consistent with the technical description document.
        - Describe test procedures used to verify the overload control mechanisms.
        - Contain data which demonstrates/indicates that the overload control mechanisms described in the technical description document have been implemented.
        - Contain details of the test set-up including the mechanisms for creating the overload. Where simulators and/or scripts are used to artificially create a load then details of these should also be included.

Expected Results:

- A technical description provides a high-level overview of the overload control design.
- A overview of the types of overload scenarios and overload control thresholds that are considered.
- Description on the types of attacks that may cause an overload to the system and how these are handled.
- A description of how the network product discards or handles input during various overload situations.
- Describes if or how the network product security functions operate and perform during overload.
- If parts of the system shutdown or take other abatement or corrective actions these should be described.

NOTE: If some of the items listed above are not applicable to a network product then, in those cases, it should be clarified by the vendor why these items are not applicable.

The test results should:

- Contain details of the overload conditions used in the test execution that are consistent with the technical description document.
- Describe the test procedures used to verify the overload control mechanisms.

- Contain data which demonstrates/indicates that the overload control mechanisms described in the technical description document have been implemented.
- Contain details of the test set-up including the mechanisms for creating the overload.

Expected format of evidence:

Documentation showing each of the points in the results sections.

4.2.3.3.4 System robustness against unexpected input.

Requirement Name: System robustness against unexpected input.

Requirement Description: During transmission of data to a system it is necessary to validate input to the network product before processing. This includes all data which is sent to the system. Examples of this are user input, values in arrays and content in protocols. The following typical implementation error shall be avoided:

- No validation on the lengths of transferred data
- Incorrect assumptions about data formats
- No validation that received data complies with the specification
- Insufficient handling of protocol errors in received data
- Insufficient restriction on recursion when parsing complex data formats
- White listing or escaping for inputs outside the values margin

Security Objective references: tba.

Test case:

This requirement will be verified by Robustness and Protocol fuzzing tests as defined in clause 4.4.4 Robustness and fuzz testing.

4.2.3.3.5 Network Product software package integrity

Requirement name: Network product Software integrity validation

Requirement reference: to be done later

Requirement Description:

1) Software package integrity shall be validated in the installation/upgrade stage.
2) Network product shall support software package integrity validation via cryptographic means, e.g. digital signature. To this end, the network product has a list of public keys or certificates of authorised software sources, and uses the keys to verify that the software update is originated from only these sources.
3) Tampered software shall not be executed or installed if integrity check fails.
4) A security mechanism is required to guarantee that only authorized individuals can initiate and deploy a software update, and modify the list mentioned in bullet 2.

Security Objective references: SOFTWARE INTEGRITY

Test case:

Test Name: TC_SW_PKG_INTEGRITY_1

Purpose:

Verify that:

1. The Network Product validates the software package integrity during the installation/upgrade stage.
2. The software package integrity validation mechanism is performed using cryptographic mechanisms, e.g. digital signature using the public keys or certificates configured in the network product.
3. Software that fails an integrity check is rejected by the network product.
4. Only authorized users are allowed to install software.

Procedure and execution steps:

Pre-Conditions:

- A network product document containing information regarding software package integrity checks, including details of how the integrity check is carried out, where public keys or certificates of sources authorised to sign software packages are stored on the network product and who these sources are, and what evidence is created to prove that the integrity check has been executed and what the result of the check was. Documentation which describes the installation procedure including how a user is authorized and authenticated to perform installation process.
- A valid network product software load/package and one that is not-valid (or could be deemed to have been tampered with) are available.

Execution Steps

The tester checks the permissions required to install software on the network product ensuring that a user is properly authenticated by the network product and that they have the required access privileges to perform the installation activity.

The tester checks, when a software package is attempted to be installed on the network product, that the software package integrity check is executed (check for evidence of execution as described in network product documentation) and that valid software is allowed to be installed but invalid software is rejected.

The tester checks the access control permissions for the software package integrity checking process, the list of public keys of authorised software sources, and any related credentials or keys for the process, to ensure that the process cannot be controlled by persons that are not authorized to do so.

Expected Results:

- Evidence that the software package integrity check has been executed for both cases of software installation (valid and invalid software packages).
- Authentication and access control mechanisms are in operation for software package installation and around the software package integrity checking mechanism.
- The installation/upgrade operation fails when using an invalid software package.
- The installation/upgrade operation is successful when using a valid software package.

Expected format of evidence:

Snapshots containing the result of the installation of package A and B.

4.2.3.4 Authentication and authorization

4.2.3.4.1 Authentication policy

4.2.3.4.1.1 System functions shall not be used without successful authentication and authorization.

Requirement Name: System functions shall not be used or accessed without successful authentication and authorization.

Requirement Description:

The usage of a system function without successful authentication on basis of the user identity and at least one authentication attribute (e.g. password, certificate) shall be prevented. System functions comprise, for example network services (like SSH, SFTP, Web services), local access via a management console, local usage of operating system and applications. This requirement shall also be applied to accounts that are only used for communication between systems. An exception to the authentication and authorization requirement are functions for public use such as those for a Web server on the Internet, via which information is made available to the public.

Security Objective references: tba.

Test case:

Test Name: TC_SYS_FUN_USAGE

Purpose:

To ensure that system functions shall not be used without successful authentication and authorization.

Procedure and execution steps:

Pre-Conditions:

1. The manufacturer shall supply the list of system functions which include network services, local access via a management console, local usage of operating system and applications.

2. The manufacturer shall supply the list of access entries for system functions.

Execution Steps

The accredited evaluator's test lab is required to execute the following steps:

1. The tester verifies, based on his/her own experience, that the list is adequate.
2. The tester verifies that the access entries to use system functions, which are listed by the manufacturer, require successful authentication on basis of the user name and at least one authentication attribute. This applies to both system functions that are locally accessible and those that are remotely accessible via a network interface.

Expected Results:

1. The network product does not allow access to any system function provided by the manufacturer without a successful user authentication.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Description of executed tests and commands
- Relevant output (e.g. Screenshot)
- Test result (Passed or not)

4.2.3.4.1.2 Accounts shall allow unambiguous identification of the user.

Requirement name: The network product shall use accounts that allow unambiguous identification of the user.

Requirement Description: Users shall be identified unambiguously by the network product. The network product shall support assignment of individual accounts per user, where a user could be a person, or, for Machine Accounts, an application, or a system. The network product shall not enable the use of group accounts or group credentials, or sharing of the same account between several users, by default. The network product shall support a minimum number of 50 individual accounts per user data base if not explicitly specified in a SCAS of a particular network product, so that accountability for each user is ensured even in large operator networks. The network product shall not support user access credentials unrelated to an account.

NOTE 1: The network product may support independent user data bases for different access methods, e.g. one data base for command shell access on OS level and another data base for GUI access. User data bases may be stored locally on the network product or on a central AAA system that the network product accesses for user authentication.

NOTE 2: This requirement does not preclude user group concepts for access control.

Security Objective references: tba.

Test case:

NOTE 3: Some typical default accounts suggest that they are shared amongst several persons (e.g. vendor_xy, support), or do not allow identification of individual users (e.g. guest, ftp, anonymous). In order to avoid overlap of this test case with clause 4.2.3.4.2.2, it is assumed for this test case that such accounts have been deleted or disabled in line with clause 4.2.3.4.2.2.

Test Name: TC_ACCOUNT_DOCUMENTATION

Purpose:

To ensure that documentation of the network product does not encourage or require the use of group accounts, group credentials, or sharing of the same account between several users. To ensure that the network product does not support credentials unrelated to an account.

Procedure and execution steps:

Pre-Conditions:

1) All user and group data bases for names and credentials supported by the network product are identified in the documentation accompanying the network product.

2) All predefined accounts and groups are identified in the documentation accompanying the Network Product.

3) Instructions of how administrator users can add accounts, groups, and credentials to the database(s) are provided in the documentation accompanying the Network Product.

4) The operations manual describes OAM user and group concepts supported by the network product.

Execution Steps:

The accredited evaluator's test lab is required to execute the following steps:

1) Review the system documentation (in particular operations manual) whether it encourages or requires the use of group accounts, group credentials, or sharing of the same account between several users.

2) Review the system documentation whether the network product requires or supports entering credentials unrelated to an account, in order to perform specific actions, e.g. to enter a "master password" for access to privileged functions.

Expected Results:

1) The reviewed documentation is in line with the requirement.

Expected format of evidence:

Test report that lists the reviewed documentation (incl. release dates and versions) and the findings.

Test Name: TC_ACCOUNT_DEFAULTS

Purpose:

To ensure that the default setup of the network product does not enable the use of group accounts or group credentials.

Procedure and execution steps:

Pre-Conditions:

1) All user and group data bases for names and credentials supported by the network product are identified in the documentation accompanying the network product.

2) Instructions of how administrator users can view all existing accounts, groups, and protected credentials in the databases are provided in the documentation accompanying the Network Product.

Execution Steps:

The accredited evaluator's test lab is required to execute the following steps:

1) After login via an account with necessary access rights (e.g. Admin) search in the databases for any group credentials. Example for Linux®: /etc/gshadow

Expected Results:

1) No group credentials are defined.

Expected format of evidence:

Test report that lists the reviewed documentation, reviewed user and group databases, and the findings.

Test Name: TC_ACCOUNT_NUMBER

Purpose:

To ensure that a minimum number of individual accounts per user data base is supported. The minimum number is defined in the requirement description of this clause.

Procedure and execution steps:

Pre-Conditions:

All user data bases for names and credentials supported by the network product are identified in the documentation accompanying the network product.

Execution Steps:

The accredited evaluator's test lab is required to execute the following steps:

Create accounts until the minimum number of accounts is reached.

Expected Results:

Successful creation of the minimum number of accounts.

Expected format of evidence:

Test report that lists the reviewed documentation, reviewed user databases, and the findings.

4.2.3.4.2 Authentication attributes

4.2.3.4.2.1 Account protection by at least one authentication attribute.

Requirement Name: Account protection by at least one authentication attribute.

Requirement Description: The various user and machine accounts on a system shall be protected from misuse. To this end, an authentication attribute is typically used, which, when combined with the user name, enables unambiguous authentication and identification of the authorized user.

Authentication attributes include:

- Cryptographic keys
- Token
- Passwords

This means that authentication based on a parameter that can be spoofed (e.g. phone numbers, public IP addresses or VPN membership) is not permitted. Exceptions are attributes that cannot be faked or spoofed by an attacker.

NOTE: Several of the above options can be combined (dual-factor authentication) to achieve a higher level of security. Whether or not this is suitable and necessary depends on the protection needs of the individual system and its data and is evaluated for individual cases.

Security Objective references: tba.

TEST CASE:

Test Name: TC_ACCOUNT_PROTECTION

Purpose:

To ensure that all accounts are protected by at least one authentication attribute.

Procedure and execution steps:

Pre-Conditions:

1) All predefined accounts are identified in the documentation accompanying the Network Product.
2) Instructions of how to create new accounts are provided in the documentation accompanying the Network Product.
3) Instructions of how administrator user can view all existing accounts in the database are provided in the documentation accompanying the Network Product.

NOTE: No test is provided here for finding undocumented hard coded accounts as such tests may be impossible to define in a general way.

Execution Steps:

The accredited evaluator's test lab is required to execute the following steps:

1) After login via account with necessary access rights (e.g. Admin) search in the database for any undocumented account.
2) Attempt login to all predefined accounts identified (either documented or not) with and without using the respective authentication attribute.
3) Create a new account by following instructions in documentation.
4) Attempt login to the newly created account.

Expected Results:

1) It is not possible to login to any predefined account without using at least one authentication attribute that satisfies the conditions in the requirement.

2) It is not possible to login to any newly created account without usage of at least one authentication attribute that satisfies the conditions in the requirement.

Expected format of evidence: tba

4.2.3.4.2.2 Predefined accounts shall be deleted or disabled.

Requirement Name: Predefined accounts shall be deleted or disabled.

Requirement Description: All predefined or default accounts shall be deleted or disabled. Many systems have default accounts (e.g. guest, ctxsys), some of which are preconfigured with or without known passwords. These standard users shall be deleted or disabled. Should this measure not be possible the accounts shall be locked for remote login. In any case disabled or locked accounts shall be configured with a complex password as specified in clause 4.2.3.4.3.1 Password Structure. This is necessary to prevent unauthorized use of such an account in case of misconfiguration.

Exceptions to this requirement to delete or disable accounts are accounts that are used only internally on the system involved and that are required for one or more applications on the system to function. Also for these accounts remote access or local login shall be forbidden to prevent abusive use by users of the system.

Security Objective references: TBA.

TEST CASE:

Test Name: TC_PREDEFINED_ACCOUNT_DELETION

Purpose:

To ensure that predefined accounts are deleted or disabled unless there is specific exception as defined in the requirement 4.2.3.4.2.2.

Procedure and execution steps:

Pre-Conditions:

1) All predefined accounts are identified in the documentation accompanying the Network Product.

2) Instructions of how administrator user can view all existing accounts in the database are provided in the documentation accompanying the Network Product.

NOTE: No test is provided here for finding undocumented hard coded accounts as such tests may be impossible to define in a general way.

Execution Steps:

1) Check in documentation of the existence of any documented predefined account and what is the reason for existence.

2) After login via account with necessary access rights (e.g. Admin) search in the database for any undocumented account.

3) Check the password complexity of such existing predefined accounts according to the test provided in clause 4.2.3.4.3.1.

4) Attempt remote login to such predefined accounts.

Expected Results:

1) Predefined accounts are either deleted/ disabled or, if existing, the reason is in accordance with the requirement exception.

2) If there are active predefined accounts in accordance with the requirement exception then there is no remote login possibility.

3) If predefined account is either disabled or locked then it shall anyway fulfil the complex password requirements as specified in clause 4.2.3.4.3.1 after enabling or unlocking it.

Expected format of evidence:

Evidence can be presented in the form of screenshot/screen-capture on showing for example a remote login failure or complexity of a password of e.g. locked or disabled accounts.

4.2.3.4.2.3 Predefined or default authentication attributes shall be deleted or disabled.

Requirement Name: Predefined or default authentication attributes shall be deleted or disabled.

Requirement Description: Predefined or default authentication attributes shall be deleted or disabled.

Normally, authentication attributes such as password or cryptographic keys will be preconfigured from producer, vendor or developer of a system. Such authentication attributes shall be changed by automatically forcing a user to change it on 1st time login to the system or the vendor provides instructions on how to manually change it.

Security Objective references: TBA.

TEST CASE:

Test Name: TC_PREDEFINED_AUTHENTICATION_ATTRIBUTES_DELETION

Purpose:

To ensure that predefined or default authentication attributes are deleted or disabled as defined in the requirement 4.2.3.4.2.3.

Procedure and execution steps:

Pre-Conditions:

1) Instructions of how administrator user can view all existing accounts in the database are provided in the documentation accompanying the Network Product.

2) All predefined accounts and their respective predefined or default passwords are identified in the documentation accompanying the Network Product.

NOTE: No test is provided here for finding undocumented hard coded accounts as such tests may be impossible to define in a general way.

Execution Steps:

1) Check in documentation of the existence of any documented predefined account and what is the login password or if any cryptographic key for such accounts is preinstalled.

2) After login via account with necessary access rights (e.g. Admin) search in the database for any undocumented account.

3) Attempt login to such predefined accounts if existing.

Expected Results:

1) When login is attempted to any predefined account the user is automatically forced to change login password at first time login to the system.

2) If there is no automatic password change enforced then recommendation and clear instructions of how to manually change the password or how to create and reinstall a new cryptographic key exist in the documentation.

Expected format of evidence:

Evidence can be presented in the form of screenshot/screen-capture on how the network product prompts for password change at first login. Also extracts from product documentation with clear instructions of how to change any default password or cryptographic key.

4.2.3.4.3 Password policy

4.2.3.4.3.1 Password Structure

Requirement Name: Password Complexity rule

Requirement Description:

The setting by the vendor shall be such that a network product shall only accept passwords that comply with the following complexity criteria:

1) Absolute minimum length of 8 characters (shorter lengths shall be rejected by the network product). It shall not be possible setting this absolute minimum length to a lower value by configuration.

2) Comprising at least three of the following categories:

    - at least 1 uppercase character (A-Z)

    - at least 1 lowercase character (a-z)

    - at least 1 digit (0-9)

    - at least 1 special character (e.g. @;!$.)

The network product shall use a default minimum length of 10 characters. The minimum length of characters in the passwords shall be configurable by the operator. The default minimum length is the value configured by the vendor before any operator-specific configuration has been applied. The special characters may be categorized in sets according to their Unicode category.

The network product shall at least support passwords of a length of 64 characters or a length greater than 64 characters.

If a central system is used for user authentication, password policy is performed on the central system and additional assurance shall be provided that the central system enforces the same password complexity rules as laid down for the local system in this subclause. If a central system is not used for user authentication, the assurance on password complexity rules shall be performed on the Network Product.

When a user is changing a password or entering a new password, the system checks and ensures that it meets the password requirements. Above requirements shall be applicable for all passwords used (e.g. application-level, OS-level, etc.).

Security Objective references: Hardening.

Test case:

Test Name: TC_PASSWORD_STRUCT

Purpose:

To verify that password structure adheres to the password complexity criteria.

To verify that password structure is configurable as per the complexity criteria.

Procedure and execution steps:

Pre-Conditions:

1. Tester has rights to create user account.

Execution Steps

Execute the following steps:

A. Test Case 1

   1. The tester logs into Network Product application using admin account.

   2. The tester creates user A following the password complexity criteria.

   3. The tester logs in as user A and attempts to change their password which contains characters from all four categories mentioned in the password complexity criteria.

B. Test Case 2

   1. The tester logins with privileged account.

   2. The tester modifies password structure policy on the network product by strengthening the policy (e.g. changing the minimum password length to 8+x, changing the minimum number of character Unicode categories to 4).

   3. The tester logs in as user A and attempts to change their password to a password with a strength of less than that permitted by the policy strengthened in step 2 above.

Expected Results:

Tester can change password only if new password fulfil the password complexity criteria

Expected format of evidence:

Evidence suitable for the interface, e.g. screenshot containing the operation result or report in text form.

4.2.3.4.3.2 Password changes

Requirement Name: tba

Requirement Description:

If a password is used as an authentication attribute, then the system shall offer a function that enables a user to change his password at any time. When an external centralized system for user authentication is used it is possible to redirect or implement this function on this system.

Password change shall be enforced after initial login.

The system shall enforce password change based on password management policy. In particular, the system shall enforce password expiry.

Previously used passwords shall not be allowed up to a certain number (Password History).
The number of disallowed previously used passwords shall be:

- Configurable;

- Greater than 0;

- And its default value shall be 3. This means that the network product shall store at least the three previously set passwords. The maximum number of passwords that the network product can store for each user is up to the manufacturer.

When a password is about to expire a password expiry notification shall be provided to the user.

Above requirements shall be applicable for all passwords used(e.g. application-level, OS-level, etc.). An exception to this requirement is machine accounts.

Security Objective references: tba.

Test case:

Test Name: TC_PASSWORD_CHANGES

Purpose:

- To check whether the network product is provisioned with the functionality that enables its user to change the password at any time.
- The network product enforces password change after initial login.
- To verify the new password adheres to the password management policy and also to verify whether it has password expiry rule.
- The network product is configured to disallow specified number of previously used passwords (Password History).

Procedure and execution steps:

Pre-Conditions:

1. Tester has account with username and password in the network product.
2. Network product vendor will provide documentation for password management policy which should include details on how to change the password, configure password expiry rule and disallowing specified number of previously used passwords.
3. The network product vendor shall supply information on how many passwords the network product can store for each user in the password history.
4. The tester has privilege to modify the number of disallowed previously used password.

Execution Steps

Execute the following steps:

A. Positive Test

Case 1:

Test case to enforce password change after initial login is covered in clause 4.2.3.4.2.3.

Case 2:

1 The tester logs into network product application using a privileged account .

2 The network product application generates password expiry notification for user Y to force user Y to change the password.

3 The tester logs out as a privileged user and logs on as user Y.

4. The tester is prompted to change his password and creates a new password by following the password policy management.

5 The network product application confirms change in password by, for example, displaying "Password Changed Successfully".

6 The tester successfully logs-in the network product application as user Y using the new password.

Case 3:

1. The tester logs into network product application using a privileged account.
2. Tester configures the network product application for number of disallowed previously used passwords to x
3. The tester requests for a password change for user Y.
4. The tester logs out of the privileged account and logs on as user Y
5. The tester creates a new password by following the password policy management.

6. If the password is not equal to any of the x previously used passwords, the network product application still accepts the new password and displays "Password Changed Successfully".

B. Negative Test

Case 1:

Test case to enforce password change after initial login is covered in clause 4.2.3.4.2.3.

Case 2:

No negative test case for this scenario.

Case 3:

1. The tester logs into network product application using privileged account.
2. Tester configures the network product application for number of disallowed previously used passwords to x for user Y.
3. The tester logs out of the privileged account and logs in as user Y
4. The tester requests for a password change.
5. The tester sets the new password to a value that is among the last x passwords used previously x times.

Expected Results:

A. Positive Test

Case 1:

Expected result for enforcing password change after initial login is covered in clause 4.2.3.4.2.3.

Case 2:

Tester can successfully change the password.

Case 3:

Tester can successfully change the password.

B. Negative Test

If the negative test case passes, this shows that network product application does not work properly and it violates the requirement.

Case 1:

Expected result for enforcing password change after initial login is covered in clause 4.2.3.4.2.3.

Case 2:

No negative test case for this scenario.

Case 3:

The tester cannot successfully change the password.

Expected format of evidence:

Evidence suitable for the interface, e.g. screenshot contains the operation result.

4.2.3.4.3.3 Protection against brute force and dictionary attacks

Requirement Name: Protection against brute force and dictionary attacks

Requirement Description:

If a password is used as an authentication attribute, a protection against brute force and dictionary attacks that hinder password guessing shall be implemented.

Brute force and dictionary attacks aim to use automated guessing to ascertain passwords for user and machine accounts. Various measures or a combination of these measures can be taken to prevent this.

The most commonly used protection measures are:

1) Using the timer delay (this delay could be the same or increased depending the operator's policy for each attempt, e.g. double the delay, or 5 minutes delay, or 10 minutes delay) for each newly entered password input following an incorrect entry ("tar pit").

2) Blocking an account following a specified number of incorrect attempts, refer to 4.2.3.4.5. However it has to be taken into account that this solution needs a process for unlocking and an attacker can force this to deactivate accounts and make them unusable.

3) Using CAPTCHA to prevent automated attempts (often used for Web applications).

4) Using a password blacklist to prevent vulnerable passwords.

NOTE 1: Password management and blacklist configuration may be done in a separate node that is different to the node under test, e.g. a SSO server or any other central credential manager.

In order to achieve higher security, it is often meaningful to combine two or more of the measures named here. It is left to the vendor to select appropriate measures.

Above requirements shall be applicable for all passwords used (e.g. application-level, OS-level, etc.). An exception to this requirement is machine accounts.

Security Objective references: tba.

Test case:

Test Name: TC_PROTECT_AGAINST_BRUTE_FORCE_AND_DICTIONARY_ATTACKS

This test applies only when the most commonly used protection measures used in the requirement are implemented. If they are not implemented, then the vendor documentation needs to provide alternative measures and the tester needs to develop suitable tests for these alternative measures. Since a vendor is free to select appropriate measures, only the vendor selected measures are to be tested.

Purpose:

To ensure that the system uses a mechanism with adequate protection against brute force and dictionary attacks

To check whether system follows commonly used preventive measures which are mentioned below.

1. Using the timer delay (e.g. doubling wait times after every incorrect attempt, or 5 minutes delay, or 10 minutes delay) after each incorrect password input ("tar pit").

2. Blocking an account following a specified number of incorrect attempts (typically 5). However administrator has to keep in account that this solution needs a process for unlocking and an attacker can utilize this process to deactivate the accounts and make them unusable.

3. Using CAPTCHA to prevent automated attempts (often used for Web interface).

4. Using a password blacklist to prevent vulnerable passwords.

Procedure and execution steps:

Pre-Conditions:

1. The password policy management of the network product is configured to use the timer delay after each incorrect password input.

2. The password policy management is configured to block an account following a specified number of incorrect password attempts (typically 5).

3. The web interface should be configured with CAPTCHA feature to prevent automated attempts.
4. CAPTCHA feature is optional and test is done only if implemented.
5. A password blacklist is configured by the tester. At least one blacklisted password (a password that meets the complexity criteria but is blacklisted) is documented.

NOTE 2: Password management and blacklist configuration may be done in a separate node that is different to the node under test, e.g. a SSO server or any other central credential manager.

6. Tester has valid credentials as an authorized user.

Execution Steps

Execute the following steps:

A. Positive Test

Case 1:

Test case to use the timer delay after each incorrect password input is covered in clause 4.2.3.4.5.

Case 2:

Test case to block an account following a specified number of incorrect attempts is covered in clause 4.2.3.4.5.

Case 3:

1. The network product's login web interface is configured with CAPTCHA feature.
2. Tester enters the valid password and correct CAPTCHA
3. Tester can successfully log into the network product.

In some cases the network product class can have two or more of the attack prevention methods available, which are a combination of Cases 1-3. In such cases the tester will need to run a combination of these test cases.

B. Negative Test

Case 1:

Test case to use the timer delay after each incorrect password input is covered in clause 4.2.3.4.5.

Case 2:

Test case to block an account following a specified number of incorrect attempts is covered in clause 4.2.3.4.5.

Case 3:

1. The network product's login web interface is configured with CAPTCHA feature.
2. Tester enters the valid password without CAPTCHA.

Case 4:

1. The tester tries to change the password to the blacklisted password.

Expected Results:

A. Positive Test

Case 1:

Expected result for the test case to use the timer delay after each incorrect password input is covered in clause 4.2.3.4.5.

Case 2:

Expected result for the test case to block an account following a specified number of incorrect attempts is covered in clause 4.2.3.4.5.

Case 3:

Tester can login only after entering the correct password and CAPTCHA.

B. Negative Test

Case 1:

Expected result for the use the timer delay after each incorrect password input is covered in clause 4.2.3.4.5.

Case 2:

Expected result for the test case to block an account following a specified number of incorrect attempts is covered in clause 4.2.3.4.5.

Case 3:

Tester cannot successfully log in to the network product.

Case 4:

Tester cannot successfully change the password to the blacklisted password.

Expected format of evidence:

Evidence suitable for the interface, e.g. screenshot containing the operation result.

4.2.3.4.3.4 Hiding password display

Requirement Name: tba

Requirement Description:

The password shall not be displayed in such a way that it could be seen and misused by a casual local observer. Typically, the individual characters of the password are replaced by a character such as "*". Under certain circumstances it may be permissible for an individual character to be displayed briefly during input. Such a function is used, for ex ample, on smartphones to make input easier. However, the entire password is never output to the display in plaintext.

Above requirements shall be applicable for all passwords used(e.g. application-level, OS-level, etc.). An exception to this requirement is machine accounts.

Security Objective references: tba.

Test case:

Test Name: TC_HIDING_PASSWORD_DISPLAY

Purpose:

Verify that the given password is not visible to the casual local observer.

Procedure and execution steps:

Pre-Conditions:

Tester has account with username and password in the network product.

Execution Steps

Execute the following steps:

1. The network product will display the login screen.

2. The tester enters the username.
3. The tester enters the password.

Expected Results:

The password shall not be displayed in such a way that it could be seen and misused by a casual local observer. Typically, the individual characters of the password are replaced by a character such as "*". Under certain circumstances it may be permissible for an individual character to be displayed briefly during input. Such a function is used, for ex ample, on smartphones to make input easier. However, the entire password is never output to the display in plaintext.

Expected format of evidence:

Evidence suitable for the interface, e.g. screenshot contains the operation results.

4.2.3.4.4 Specific Authentication use cases

4.2.3.4.4.1 Network Product Management and Maintenance interfaces

Requirement Name: Network Product Management and Maintenance interfaces

Requirement Description: The network product management shall support mutual authentication mechanisms, the mutual authentication mechanism can rely on the protocol used for the interface itself or other means.

Security Objective references: Secure Network Product Administration.

Test case:

Test Name: TC_MUTUAL_AUTHENTICATION-ON_NETWORK_PRODUCT_MANAGEMENT_PROTOCOLS

Purpose:

Verify that:

There is mutual authentication of entities for management interfaces on the network product.

Procedure and execution steps:

Pre-conditions: Documentation that lists each of the management protocols and describes the authentication mechanism used for each one.

Execution Steps

1. The tester checks that the authentication mechanisms have been configured on the network product.
2. The tester triggers communication between network product and a test entity that has a legitimate authentication credential.
3. Then, the tester triggers communication between network product and a test entity that doesn't have a legitimate authentication credential.

Expected results:

- Mutual authentication is successful and communication between network product and the entity with correct credentials can be established.
- Mutual authentication fails and communication between the network product and the entity with incorrect credentials cannot be established.

Expected format of evidence: Test result pass/fail recorded by tester.

4.2.3.4.5 Policy regarding consecutive failed login attempts

Requirement Name: tba

Requirement Description:

a) The maximum permissible number of consecutive failed user account login attempts should be configurable by the operator. The definition of the default value set at manufacturing time for maximum number of failed user account login attempts shall be less than or equal to 8, typically 5. After the maximum permissible number of consecutive failed user account login attempts is exceeded by a user there shall be a block delay in allowing the user to attempt login again. This block delay and also the capability to set period of the block delay, e.g. double the delay, or 5 minutes delay, or 10 minutes delay, after each login failure should be configurable by the operator. The default value set at manufacturing time for this delay shall be greater than or equal to 5 sec.

b) If supported, infinite (permanent) locking of an account that has exceeded maximum permissible number of consecutive failed user account login attempts should also be possible via configuration, with the exception of administrative accounts which shall get only temporarily locked.

Security Objective references: tba.

TEST CASE*:*

Test Name: TC_FAILED_LOGIN_ATTEMPTS

Purpose:

To ensure that the policy regarding failed login attempts is adhered to.

Case 1: Testing for requirement 4.2.3.4.5 a)

Procedure and execution steps:

Pre-Conditions:

1) At least one user account has been created as per manufacturer's instructions.

2) Directions of how to configure the maximum permissible number of consecutive failed user account login attempts and the default value of this number are identified in the documentation accompanying the Network Product. Default value shall be stated as well.

3) Directions of how to configure the block delay in allowing a user attempt to login again when the number of failed login attempts has exceeded the maximum number are identified in the documentation accompanying the Network Product. Default value of the delay shall be stated as well.

Execution Steps:

The accredited evaluator's test lab is required to execute the following steps:

1) Check default values from precondition 2 and 3.

2) Perform consecutive failed login attempts for the user account until the default maximum number of precondition 2 is reached.

3) Attempt again one extra login, which fails again.

4) Attempt one extra login in less time than the default for the delay of precondition 3, using the correct credentials.

5) Attempt one extra login in more time than the default for the delay of precondition 3, using the correct credentials.

Expected Results:

1) Default values from precondition 2 and 3 are in accordance with the requirement.

2) In execution step 4, the login attempt shall be rejected in all cases.

3) In execution step 5, the login attempt shall be accepted.

4) In execution step 6, it is verified that the user can login only at the last login attempt.

Expected format of evidence: tba

Case 2: Testing for requirement 4.2.3.4.5 b)

Procedure and execution steps:

Pre-Conditions:

1. At least one user account has been created as per manufacturer's instructions.

2. Directions of how to configure the maximum permissible number of consecutive failed user account login attempts and the default value of this number are identified in the documentation accompanying the Network Product. Default value shall be stated as well.

3. Directions of how to optionally configure permanent locking for non-administrative accounts shall be stated as well.

Execution Steps:

The accredited evaluator's tes**t** lab is required to execute the following steps:

1. Check default values from precondition 2.

2. Perform consecutive failed login attempts for the user account until the default maximum number of precondition 2 is reached.

3. Attempt again one extra login, which fails again.

4. Attempt one extra login in more time than the default for the delay of precondition 3, using the correct credentials.

5a. If supported enable permanent locking of accounts exceeding the maximum permissible number of consecutive failed user account login attempts and repeat steps 1-4 for a normal user.

5b. If supported enable permanent locking of accounts exceeding the maximum permissible number of consecutive failed user account login attempts and repeat steps 1-4 for a user with administrative access rights.

Expected Results:

In execution step 5a it is verified that the user cannot login at any execution step.

In execution step 5b it is verified that an administrator user can successfully login only at execution step 5b.

Expected format of evidence: tba

4.2.3.4.6 Authorization and access control

4.2.3.4.6.1 Authorization policy

Requirement Name: tba

Requirement Description:

The authorizations for accounts and applications shall be reduced to the minimum required for the tasks they have to perform.

Authorizations to a system shall be restricted to a level in which a user can only access data and use functions that he needs in the course of his work. Suitable authorizations shall also be assigned for access to files that are components of the operating system or of applications or that are generated by the same (e.g. configuration and logging files).

Alongside access to data, execution of applications and components shall also take place with rights that are as low as possible. Applications should not be executed with administrator or system rights.

Security Objective references: tba.

Test case: verify authorization policy is in place and that user access and data access in the system are according to the authorization policy.

Procedure and execution steps:

Pre-Conditions:

Documentation describing the authorization policy defined for the system including details on the lowest access rights assigned to user accounts, access to data, application execution and components.

Execution Steps:

1. Assign access rights (e.g. read only) to user accounts, data files, and applications.
2. Operations, that are allowed as per authorization policy (as defined in the network product documentation), are attempted via the different user accounts, data files, and applications.

Expected Results:

1. User accounts, data files, and applications are allowed to be accessed (e.g. able to read but not write to a file, able to execute an application as a user account without administrator rights, etc.) according to the access rights assigned.
2. User accounts, data files, and applications are not allowed to be accessed above the access rights assigned (e.g. able to write to a read only file, able to execute an application as an administrator, etc.).

Expected format of evidence:

Pass/fail results as recorded by the tester.

4.2.3.4.6.2 Role-based access control

Requirement Name: tba

Requirement Description:

The network product shall support Role Based Access Control (RBAC). A role-based access control system uses a set of controls which determines how users interact with domains and resources. The domains could be Fault Management (FM), Performance Management (PM), System Admin, etc. The RBAC system controls how users or groups of users are allowed access to the various domains and what type of operation they can perform, i.e. the specific operation command or command group (e.g. View, Modify, Execute).

The network product supports RBAC, in particular, for OAM privilege management for network product Management and Maintenance, including authorization of the operation for configuration data and software via the network product console interface.

Security Objective references: tba.

Test case:

Purpose:

Verify that users are granted access with role-based privileges.

Procedure and execution steps:

Pre-Conditions:

Documentation describing the role based access control system including details on which user roles are defined.

Execution Steps

1. User accounts which are assigned to different access roles are created.
2. Operations, that are allowed by different roles (as defined in the network product documentation), are attempted via the different user accounts.

Expected Results:

1. Users that are assigned to a role that is not allowed to execute an operation are prevented from executing the operation.
2. Users that are assigned to a role that is allowed to execute an operation can successfully execute the operation.

Expected format of evidence:

Pass/fail results as recorded by the tester.

4.2.3.5 Protecting sessions

4.2.3.5.1 Protecting sessions – logout function

Requirement Name: Protecting sessions – logout function

Requirement Description: The system shall have a function that allows a signed in user to logout at any time. All processes under the logged in user ID shall be terminated on log out. The network product shall be able to continue to operate without interactive sessions.

Only for debugging purposes, processes under a logged in user ID may be allowed to continue to run after detaching the interactive session.

Security Objective references: tba.

Test Name: TC_PROTECTING_SESSION_LOGOUT

Purpose:

To ensure a signed in user can logout at any time.

Procedure and execution steps:

Pre-Conditions:

- The manufacturer shall declare that it has a function that allows a signed in user to logout at any time.
- The tester has privileges to create a new account or use an existing account.

Execution Steps:

The accredited evaluator's test lab is required to execute the following steps:

1) The tester creates a new account.

2) The tester uses the new account or an existing account to log into network product. After x minutes the tester tries to logout network product.

NOTE: The value of x can be arbitrarily set by the tester.

Expected Results:

- The tester can use a new account or an existing account to log into network product and logout network product after x minutes.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Settings, and configurations used
- Test result (Passed or not)

4.2.3.5.2 Protecting sessions – Inactivity timeout

Requirement Name: Protecting sessions – inactivity timeout

Requirement Description: An OAM user interactive session shall be terminated automatically after a specified period of inactivity. It shall be possible to configure an inactivity time-out period.

NOTE: The kind of activity required to reset the timeout timer depends on the type of user session.

Test Name: TC_PROTECTING_SESSION_ INAC TIMEOUT

Purpose:

To ensure an OAM user interactive session shall be terminated at inactivity timeout.

Procedure and execution steps:

Pre-Conditions:

- The tester has privileges to create an OAM user interactive session.
- The tester has privileges to configure the inactivity time-out period for user interactive session.
- Session log should be enabled.

Execution Steps

1. The tester creates OAM user A interaction session.
2. The tester configures the inactivity time-out period for user A to x minute, for example 1 minute.
3. The tester does not make any actions on the network production in x minutes. After that, the tester checks whether OAM user A interaction session has been terminated automatically.

Expected Results:

- In step 3, OAM user A interaction session has been terminated automatically after x minute.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Session log

- Settings, protocols and configurations used
- Test result (Passed or not)

Security Objective references: tba.

4.2.3.6 Logging

4.2.3.6.1 Security event logging

Requirement Name: Security event logging

Requirement Description: Security events shall be logged together with a unique system reference (e.g. host name, IP or MAC address) and the exact time the incident occurred. For each security event, the log entry shall include user name and/or timestamp and/or performed action and/or result and/or length of session and/or values exceeded and/or value reached.

IETF RFC 3871, section 2.11.10 specifies the minimum set of security events. Each vendor shall document what security events the product logs so that it can be verified by testing.

In particular, it shall be possible to log the following events (which are intended to be supported by the network product and which can be enabled by default at manufacturing time or at a later time by the operator):

EventTypes	Description	Event data to be logged
Incorrect login attempts	Records any user incorrect login attempts to the network product	• Username, • Source (IP address) if remote access • Timestamp
Administrator access	Records any access attempts to accounts that have system privileges.	• Username, • Timestamp, • Length of session, • Source (IP address) if remote access
Account administration	Records all account administration activity, i.e. configure, delete, enable, and disable.	• Administrator username, • Administered account, • Activity performed (configure, delete, enable and disable) • Timestamp
Resource Usage	Records events that have been triggered when system parameter values such as disk space, CPU load over a longer period have exceeded their defined thresholds.	• Value exceeded, • Value reached (Here suitable threshold values shall be defined depending on the individual system.) • Timestamp
Configuration change	Changes to configuration of the network device	• Change made • Username
Reboot/shutdown/crash	This event records any action on the network device that forces a reboot or shutdown OR where the network device has crashed.	• Action performed (reboot, shutdown, etc.) • Username (for intentional actions) • Timestamp
Interface status change	Change to the status of interfaces on the network device (e.g. shutdown)	• Interface name and type • Status (shutdown, missing link, etc.) • Timestamp

In addition, optionally it shall be possible to log also the following event (if supported):

EventTypes	Description	Event data to be logged
Change of group membership or accounts	Any change of group membership for accounts	• Administrator username, • Administered account, • Activity performed (group added or removed) • Timestamp.

Security Objective references: tba.

Test case:

Test Name: TC_SECURITY_EVENT_LOGGING

Purpose:

To verify that the network product correctly logs all required security event types.

Procedure and execution steps:

Pre-Conditions:

- The following information shall be provided by the documentation accompanying the network product:
  - The log where the event is recorded and how it can be accessed (e.g. the complete path).
  - If the event type is enabled by default or how to enable it.
  - What O&M services can be used on the Network Product in the configuration according to the pre-requisites for testing in clause 4.1 and how to use them.
- The tester has the needed administrative privileges to sufficiently perform the tests
- If needed for testing specific O&M services, a tester machine is available.

Execution Steps

For each O&M service perform the following test steps

- The Tester sequentially triggers each security event listed in the requirement, while covering each option detailed in the individual security event descriptions.
- The Tester verifies whether the security events, and their individual options, were correctly logged. In particular it is verified whether they include at least the event data specified as required to be logged.

Expected Results:

All security events are appropriately logged, including all required event data.

Expected format of evidence:

The testing report contains the following information for each security event:

- List of O&M services
- Commands executed per O&M services
- The relevant parts of the logs in appropriate form (e.g. file, screenshot)
- Test result (Passed or not)

4.2.3.6.2 Log transfer to centralized storage

Requirement Name: Log transfer to centralized storage

Requirement Description:

a) The Network Product shall support forwarding of security event logging data to an external system. Secure transport protocols in accordance with clause 4.2.3.2.4, shall be used.

b) Log functions should support secure uploading of log files to a central location or to an external system for the Network Product that is logging.

Security Objective references: tba.

Test Name: TC_LOG TRANS_TO_CENTR STORAGE

Purpose:

To ensure log shall be transferred to centralized storage.

Procedure and execution steps:

Pre-Conditions:

- The manufacturer shall list the standard protocols which transfer security event logging data.
- The session between network product and central location or external system for network product log functions has been set up.
- The tester has privilege to operate network product and related logs can be outputted.

Execution Steps

1. The tester configures the network product to forward event logs to an external system (according to bullet a) of requirement) and related logs are sent out.
2. The tester checks whether the used transport protocol is secure protocol.
3. The tester checks whether the central location or external system for network product log functions has stored the related logs.
4. The tester configures the network product for secure upload of event log files to an external system (according to bullet b) of requirement) and performs a log file upload.
5. The tester checks whether the used transport protocol for log file upload is a secure standard protocol.
6. The tester checks whether the central location or external system for network product log functions has stored the related logs.

Expected Results:

- The listed transport protocols are secure protocols.
- The used transport protocol for log file upload is a secure standard protocol.
- The tester finds that the central location or external system for network product log functions has stored the related logs.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Settings, protocols and configurations used,
- Screenshot
- Test result (Passed or not)

4.2.3.6.3 Protection of security event log files

Requirement Name: Protection of security event log files

Requirement Description: The security event log shall be access controlled (file access rights) so only privileged users have access to the log files.

Security Objective references: tba.

Test case:

Purpose:

Verify that the log(s) is(are) only accessible by privileged user(s).

Procedure and execution steps:

Pre-Conditions:

- Documentation describing where logs are stored and how these logs are accessed and the Network Product interfaces that these logs can be access from.

Execution Steps

1. The tester attempts to access log files using users accounts with and without the correct permissions for accessing log files.

2. Repeat the test as described in step 1 using each of the interfaces as described in the Network Product documentation.

Expected Results:

The tester checks that log files are accessible when a user with the appropriate authorisation attempts to access them and fails when a user without the correct permissions attempts to access them

Expected format of evidence:

Pass/fail result as recorded by the tester.

4.2.4 Operating systems

4.2.4.1 General operating system requirements and related test cases

4.2.4.1.1 Availability and Integrity

4.2.4.1.1.1 Handling of growing content

Requirement Name: Growing (dynamic) content shall not influence system functions.

Requirement Description:

Growing or dynamic content (e.g. log files, uploads) shall not influence system functions. A file system that reaches its maximum capacity shall not stop a system from operating properly. Therefore, countermeasures shall be taken such as usage of dedicated filesystems, separated from main system functions, or quotas, or at least a file system monitoring to ensure that this scenario is avoided.

Test Case:

Test Name: TC_HANDLING_OF_GROWING_CONTENT

Purpose:

To verify that the growing or dynamic content does not influence system functions.

Procedure and execution steps:

Pre-Conditions:

1. Growing or dynamic content sources like e.g. log files and their paths are documented.

2. Measures that are taken to protect system functions from growing or dynamic content that may exhaust file system capacity are documented.

3. All logging capabilities that are not enabled by default are enabled manually as per the documentation instructions.

Execution Steps

1. Tester checks that the sources that are susceptible to being exhausted have been documented and measures aimed to counteract this are described.

2. Tester enables monitoring of the system operation.

3. Tester initiates traffic that causes increase of log files and monitors the system behaviour until the log file either reaches its quota or until file system is exhausted.

4. In case file uploading is allowed (e.g. via SFTP) the tester initiates file uploading and tries to exhaust the file system.

Expected Results:

1. It is verified that the taken measures are sufficient so that system operation is not influenced by growing or dynamic content at any case.

Expected format of evidence:

System monitoring data (e.g. Alarms, logs, CPU utilization, etc.).

4.2.4.1.1.2 Handling of ICMP

Requirement Name: Processing of ICMPv4 and ICMPv6 packets

Requirement Description:

Processing of ICMPv4 and ICMPv6 packets which are not required for operation shall be disabled on the network product. In particular, there are certain types of ICMP4 and ICMPv6 that are not used in most networks, but represent a risk.

ICMP message types which on receipt lead to responses or to configuration changes are not mentioned in this requirement, but they may be necessary to support relevant and specified networking features. Those must be documented.

Certain ICMP types are generally permitted and do not need to be specifically documented. Those are marked as "Permitted" in below table.

The network product shall not send certain ICMP types by default, but it may support the option to enable utilization of these types (e.g. for debugging). This is marked as "Optional" in below table.

Type (IPv4)	Type (IPv6)	Description	Send	Respond to
0	128	Echo Reply	Optional (i.e. as automatic reply to "Echo Request")	N/A
3	1	Destination Unreachable	Permitted	N/A
8	129	Echo Request	Permitted	Optional
11	3	Time Exceeded	Optional	N/A
12	4	Parameter Problem	Permitted	N/A
N/A	2	Packet Too Big	Permitted	N/A
N/A	135	Neigbor Solicitation	Permitted	Permitted
N/A	136	Neighbor Advertisement	Permitted	N/A

The network product shall not respond to, or process (i.e. do changes to configuration), under any circumstances certain ICMP message types as marked in below table.

Type (IPv4)	Type (IPv6)	Description	Send	Respond to	Process (i.e. do changes to configuration)
5	137	Redirect	N/A	N/A	Not Permitted
13	N/A	Timestamp	N/A	Not Permitted	N/A
14	N/A	Timestamp Reply	Not Permitted (i.e. as automatic reply to "Timestamp")	N/A	N/A
N/A	133	Router Solicitation	N/A	Not Permitted	Not Permitted
N/A	134	Router Advertisement	N/A	N/A	Not Permitted

Test Case:

The test for this requirement can be carried out using a suitable tool or manually by performing the steps described below. If a tool is used then the tester needs to provide evidence, e.g. by referring to the documentation of the tool, that the tool actually provides functionality equivalent to the steps described below.

Test Name: TC_HANDLING_OF_ICMP

Purpose:

To verify that the network product does not reply to certain ICMP types in accordance with the requirement. To verify that the network product does not send 'Time Exceeded'.

To verify that the network product does not process the following ICMPv4 and ICMPv6 types:

- "Redirect (5)"
- Router Solicitation
- Router Advertisement

Procedure and execution steps:

Pre-Conditions:

- The tester knows whether the network product supports IPv4 and/or IPv6:
- If applicable, the tester has the needed system privileges for confirming that the ICMP messages with types "Not Permitted" to process are indeed not leading to configuration changes..
- If applicable, the tester has the needed system privileges for confirming that certain ICMP message types are dropped by the network product on receipt.
- A tester machine is available and equipped with a suitable ICMP packets generator tool.

Execution Steps

The following needs to be done for all IP protocol versions (IPv4 and/or IPv6) supported by the network element.

For verifying that the network product does not reply to ICMP messages with types where this is not permitted: The tester sends samples of the applicable ICMP messages from the tester machine to the network product and verifies by appropriate means that

- the messages are dropped on receipt by the network product (e.g. by means of appropriate firewall rules),
- or no response is sent out towards the test machine,
- or there are other means ensuring that the ICMP messages cannot trigger a response.

For verifying that the network product does not change its configuration due to receiving ICMP messages with types where this is not permitted: The tester sends samples of the applicable ICMP messages from the tester machine to the network product and verifies by appropriate means that

- the messages are dropped on receipt by the network product (e.g. by means of appropriate firewall rules),
- or the network product's applicable system configuration remains unchanged upon receipt of the messages,
- or there are other means ensuring that the ICMP messages cannot lead to configuration changes.

The tester utilizes appropriate means to verify consistency between the documentation in regard to ICMP and the network product.

Expected Results:

The ICMP messages which are "Not Permitted" to generate a response from the network product do not generate a response.

The ICMP messages which are "Not Permitted" to change the configuration of the network element do not change the configuration.

ICMP message types which lead to responses or to configuration changes on receipt, if neither mentioned in the requirement nor in the documentation, are not enabled.

Expected format of evidence:

The following information needs to be retained and included into the report as appropriate:

- Tools used and their configuration
- Tool output
- Test result (Passed or not)

4.2.4.1.1.3 Handling of IP options and extensions

Requirement Name: IP packets with unnecessary options or extension headers shall not be processed.

Requirement Description:

IP packets with unnecessary options or extension headers shall not be processed. IP options and extension headers (e.g. source routing) are only required in exceptional cases. So, all packets with enabled IP options or extension headers shall be filtered.

Test Case:

The test for this requirement can be carried out using a suitable tool or manually by performing the steps described below. If a tool is used then the tester needs to provide evidence, e.g. by referring to the documentation of the tool, that the tool actually provides functionality equivalent to the steps described below.

Test Name: TC_HANDLING-IP-OPTIONS-AND-EXTENSIONS

Purpose: To verify that the network product provides functionality to filter out IP packets with unnecessary options or extension headers.

Procedure and execution steps:

Pre-Conditions:

- The manufacturer declares in the documentation accompanying the network product at least the following information:
  - The support of filtering capability for IP packets with unnecessary options or extensions headers.
  - The actions performed by the network product when an IP packet with unnecessary options or extensions headers is received (e.g. the packet is dropped, the options or extensions are ignored and the packet is treated as if it has no IP options, etc.) .
  - Guidelines on how to enable and configure this filtering capability.

- The network product has at least one physical interface named if1 supporting both IPv4 and IPv6. If the network product does not support IPv6 then IPv6 related steps and checks may be skipped**.**

- A network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product is available .

- The tester has administrative privileges.

- A tester machine is available with a tool able to send IPv4 packets with the IP Options and IPv6 packets (if supported by the network product) with Extension Header set (e.g. Scapy).

Execution Steps

1. The tester logs in the network product.

2. The tester configures on the network product a filtering rule to drop all IP packets containing an IP Option set

   a) The tester establishes an O&M session on if1 interface

   b) Using the tool (e.g. Scapy) the tester sends from the tester machine an IPv4 TCP SYN packet with an appropriate destination portto if1 interface without setting any IP Options

   c) Using the network traffic analyser, the tester verifies that the IP packet is received by the network product and the tester verifies that the corresponding ACK message is sent back.

   d) Using the tool (e.g. Scapy) the tester sends an IPv4 TCP SYN packet with an appropriate destination port and an IP Option set to the if1 interface

   e) Using the network traffic analyser, the tester verifies that the IP packet is received by the network product but no ACK message is sent back. This confirms the packet is dropped as expected from the filtering rule.

3. The tester configures on the network product a filtering rule to drop all incoming packets based on specific Extension Header Types, e.g. packets with the Routing Header extension. Step 3 may be skipped if the network product does not support IPv6.

   a) Using the tool (e.g. Scapy) the tester sends from the tester machine an IPv6 TCP SYN packet with an appropriate destination port to if1 interface without setting any extension header

   b) Using the network traffic analyser, the tester verifies that the IP packet is received by the network product and the tester verifies that the corresponding ACK message is sent back.

   c) Using the tool (e.g. Scapy) the tester sends an IPv6 TCP SYN packet with an appropriate destination port and an extension header set to the if1 interface

   d) Using the network traffic analyser, the tester verifies that the IP packet is received by the network product but no ACK message is sent back. This confirms the packet is dropped as expected from the filtering rule.

Expected Results:

The network product discards IPv4 packets with unnecessary options or IPv6 packets (assuming the network product supports IPv6) with extension header.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Used tools and their configurations

- Settings and configurations used

- Pcap trace

- Screenshot

- Test result (Passed or not)

4.2.4.1.2 Authentication and Authorization

4.2.4.1.2.1 Authenticated Privilege Escalation only

Requirement Name: There shall not be a privilege escalation method in interactive sessions (CLI or GUI) which allows a user to gain administrator/root privileges from another user account without re-authentication.

Requirement Description:

There shall not be a privilege escalation method in interactive sessions (CLI or GUI) which allows a user to gain administrator/root privileges from another user account without re-authentication.. Implementation example: Disable insecure privilege escalation methods so that users are required to (re-)login directly into the account with the required permissions.

Test Case:

Test Name: TC_OS_PRIVILEGE

Purpose:

To ensure that privileged operating system functions shall not be used without successful authentication and authorization, and that violations of this requirement are documented and strictly limited in number and functionality.

Procedure and execution steps:

Pre-Conditions:

1. The manufacturer shall provide documentation of the operating system(s) used in the network product.
2. The manufacturer shall supply a list "A" of operating system functions which a system user can use to explicitly gain higher privileges, and how these functions are configured. Unix® example: sudo command and its configuration file /etc/sudoers.
3. The manufacturer shall supply a list "B" of operating system commands, GUI functions, and files which will execute specifically limited tasks automatically with higher privileges, even when used by a low-privileged user. List "B" shall also contain:
    - configuration of these commands and GUI functions;
    - owner and permission settings of files;
    - justification for having the command, GUI function or file on the network product
      Unix® example: root-owned files with SUID and SGID permissions.

Execution Steps

The accredited evaluator's test lab is required to execute the following steps:

1. The tester logs into the network product and verifies that list "A" is accurate, based on his expert knowledge of the operating system(s) used in the network product, and operating system documentation.
2. The tester verifies that entries in the list "A" require successful authentication for all users without exception, on basis of the user name and at least one authentication attribute.
3. The tester logs into the network product and verifies that list "B" is accurate, based on his expert knowledge of the operating system(s) used in the network product, and operating system documentation. Unix® example: To list files with SUID and SGID permissions, the following commands can be used:

    ```
    SUID:    find / -perm -4000 -type f -exec ls {} \; > suid_files.txt

    SGID:    find / -perm -2000 -type f -exec ls {} \; > sgid_files.txt
    ```

4. The tester verifies that file entries in the list "B" do not have write permissions for anyone else than the owner.

5. The tester verifies that entries in the list "B" only allow execution of specifically limited tasks which are needed on this network product, based on his expert knowledge of the operating system(s) used in the network product, and operating system documentation.

6. The tester logs into the network product and tests for every entry in the list "B" that it does not provide a means to execute arbitrary functions with administrator/root privileges, e.g. via a shell escape.

Expected Results:

1. The network product does not allow a user to gain administrator/root privileges from another user account without re-authentication.

2. If a network product provides functions and files which execute specifically limited tasks automatically with higher privileges, it ensures that these limits cannot be bypassed.

3. The system documentation about means for a user to gain administrator/root privileges from another user account accurately describes the network product.

Expected format of evidence:

A test report provided by the accredited evaluator's test lab which will consist of the following information:

- Documentation provided by the vendor: lists "A" and "B"
- Description of executed tests and commands
- Relevant output (e.g. screenshot or terminal log)
- Test result (passed or not passed)

4.2.4.2 UNIX® specific requirements and related test cases

4.2.4.2.1 General

NOTE: The term 'UNIX®' is throughout the present document meant to include all major UNIX®-like derivatives, including Linux®.

4.2.4.2.2 System account identification

Requirement Name: System account identification

Requirement Description: Each system account in UNIX® shall have a unique UID.

Security Objective references: tba.

Test case:

Test Name: TC_UNIQUE_SYSTEM_ACCOUNT_IDENTIFICATION

Purpose: To verify that UNIX® account UIDs are assigned uniquely.

Procedure and execution steps:

Pre-Conditions: UNIX® is used on the MME.

Execution Steps

1. Create several UNIX® accounts.

2. Check UIDs of created accounts and of existing system accounts and, in particular, the root account.

Expected Results: The UIDs are all different and, in particular, only the root account has UID = 0.

4.2.5 Web Servers

4.2.5.1 HTTPS

Requirement Name: HTTPS

Requirement Description: The communication between Web client and Web server shall be protected using TLS. The TLS profile defined in Annex E of TS 33.310 shall be followed with the following modifications:

Cipher suites with NULL encryption shall not be supported

Security Objective references: tba.

Test case:

Test Name: HTTPS

Purpose: Verify the above requirement.

Procedure and execution steps, expected results, expected format of evidence:

These are the same as for the test case in clause 4.2.3.2.4, except that, for execution step 2, it is tested that NULL encryption is not supported.

4.2.5.2 Logging

4.2.5.2.1 Webserver logging

Requirement Name: Webserver logging

Requirement Description: Access to the webserver shall be logged. The web server log shall contain the following information:

- Access timestamp
- Source (IP address)
- (Optional) Account (if known)
- (Optional) Attempted login name (if the associated account does not exist)
- Relevant fields in http request. The URL should be included whenever possible.
- Status code of web server response

Security Objective references: tba.

Test case:

Test Name: **TC_WEBSERVER_LOGGING**

Purpose:

Verify that all accesses to the webserver are logged with the required information.

Procedure and execution steps:

Pre-Condition:

Network Product documentation which contains information on log file location and procedure to access it.

Tester has the necessary privileges to access the log files.

Execution Steps

Execute the following steps:

1. The tester tries to login to the webserver using the correct and incorrect login credentials.
2. The tester verifies whether the login attempts were logged correctly with all of the required information.

Expected Results:

All webserver events are logged with all of the required information.

Expected format of evidence:

Testing report contains copies of the log file showing the captured information.

4.2.5.3 HTTP User sessions

Requirement Name: User sessions

Requirement Description:

To protect user sessions the Network Product shall support the following session ID and session cookie requirements:

1. The session ID shall uniquely identify the user and distinguish the session from all other active sessions.
2. The session ID shall be unpredictable.
3. The session ID shall not contain sensitive information in clear text (e.g. account number, social security, etc.).
4. In addition to the Session Idle Timeout (see clause 4.2.3.5.2 Protecting sessions – Inactivity timeout), the Network Product shall automatically terminate sessions after a configurable maximum lifetime This maximum lifetime defines the maximum session span. When the maximum lifetime expires, the session shall be closed, the session ID shall be deleted and the user shall be forced to (re)authenticate in the web application and to establish a new session. The default value for this maximum lifetime shall be set to 8 hours.
5. Session ID's shall be regenerated for each new session (e.g. each time a user logs in).
6. The session ID shall not be reused or renewed in subsequent sessions.
7. The Network Product shall not use persistent cookies to manage sessions but only session cookies. This means that neither the "expire" nor the "max-age" attribute shall be set in the cookies.
8. Where session cookies are used the attribute 'HttpOnly' shall be set to true.
9. Where session cookies are used the 'domain' attribute shall be set to ensure that the cookie can only be sent to the specified domain.
10. Where session cookies are used the 'path' attribute shall be set to ensure that the cookie can only be sent to the specified directory or sub-directory.
11. The Network Product shall not accept session identifiers from GET/POST variables.
12. The Network Product shall be configured to only accept server generated session ID's.

Security Objective references: tba.

Test case:

Purpose:

Verify that the above 12 session ID and session cookie requirements have been met.

Procedure and execution steps:

Pre-Conditions:

- The Network Product uses a session ID that is communicated between the client and Network Product to establish and maintain a session.
- Documentation describing how a session is maintained and where the session ID is stored / and how this is communicated and after how long sessions expire.
- The documentation should describe the algorithm used to generate the session IDs.

Execution Steps

1. The tester logs in repeatedly with different user IDs and a number of times with the same user ID in a row and collects the session IDs according to the documentation and the user IDs associated with them. The tester verifies that:
   a. The session IDs are different between sessions of the same and different users;
   b. The session IDs seems random based on his/her own experience. The tester may use tests like the bitstream test or the count-the-1s-tests from the diehard test suite. The tester documents how randomness was verified;
   c. The session IDs are always different between sessions, also when the user ID is the same.
2. The tester verifies that when session cookies are used
   a. neither the "expire" or the "max-age" is set;
   b. the 'HttpOnly' is set to true;
   c. the 'domain' attribute is set to the correct domain;
   d. the 'path' attribute is set to the correct directory or sub-directory.
3. The tester verifies that it is impossible to:
   a. access a session by retrieving the session ID and communicating the session ID through a POST or GET variable.
   b. generate a session ID on the client by attempting to login with a custom generated session ID.
   c. keep a session alive for longer than the configured maximum lifetime (by default 8 hours).

Expected Results:

1. A list of session IDs and user IDs that are different between sessions even when the tester has logged in with the same user and that are unpredictable as is confirmed by the entropy calculation.
2. A confirmation from the tester that the correct variables are indeed set.
3. A denied access to the tester when attempting the two login steps of step 3 and an expired session in step 3c.

Expected format of evidence:

A confirmation that the tester has confirmed that:

1. Session IDs follow the rules 1-3, 5, 6.
2. A session times out after 8 hours or sooner according to the documentation.
3. The correct cookie settings are used.
4. The network product does not accept customly generated session IDs and that session IDs over GET or POST are ignored.

4.2.5.4 HTTP input validation

Requirement Name: Input validation

Requirement Description:

The Network Product shall have a mechanism in place to ensure that web application inputs are not vulnerable to command injection or cross-site scripting attacks. The Network Product shall validate, filter, escape, and encode user-controllable input before it is placed in output that is used as a web page that is served to other users.

Security Objective references: tba.

Test case:

This requirement is covered by the basic vulnerability testing as described in clause 4.4.

4.2.6 Network Devices

4.2.6.1 Protection of Data and Information

Refer to clause 4.2.3.2 for requirements on protection of data and information.

4.2.6.2 Protecting availability and integrity

4.2.6.2.1 Packet filtering

Requirement Name: Packet filtering

Requirement Description:

The Network Product shall provide a mechanism to filter incoming IP packets on any IP interface (see RFC 3871 for further information).

In particular the Network Product shall provide a mechanism:

1) To filter incoming IP packets on any IP interface at Network Layer .and Transport Layer of the stack ISO/OSI.

2) To allow specified actions to be taken when a filter rule matches. In particular at least the following actions should be supported:

   - Discard/Drop: the matching message is discarded, no subsequent rules are applied and no answer is sent back.

   - Accept: the matching message is accepted.

   - Account: the matching message is accounted for i.e. a counter for the rule is incremented. This action can be combined with the previous ones. This feature is useful to monitor traffic before its blocking.

3) To enable/disable for each rule the logging for Dropped packets, i.e. details on messages matching the rule for troubleshooting.

4) To filter on the basis of the value(s) of any portion of the protocol header.

5) To reset the accounting.

6) The Network Product shall provide a mechanism to disable/enable each defined rule.

Security Objective references: PROTECTED COMMUNICATIONS, HARDENING.

Test case:

Test Name: TC_PACKET_FILTERING

Purpose:

Verify that the system provides functionality for incoming packet filtering

Procedure and execution steps:

Pre-Conditions:

- The Network Product has packet filtering enabled.
- The Network Product has 2 different logical or physical Ethernet ports and each port is connected to a host**.**

Execution Steps

1. The tester configures the Network Product to only allow ICMP traffic from host 1.
2. The tester initiates ping traffic from host 1.
3. The tester initiates ping traffic from host 2.

Expected Results:

Host 1 will receive a 'ping' answer back, but host 2 will not.

Expected format of evidence:

NA

4.2.6.2.2 Interface robustness requirements

Requirement Name: Manipulated packets that are sent to an address of the network device shall not lead to an impairment of availability.

Requirement Description:

A network device shall be not affected in its availability or robustness by incoming packets, from other network element, that are manipulated or differing the norm. This means that appropriate packets shall be detected as invalid and be discarded. The process shall not be affecting the performance of the network device. This robustness shall be just as effective for a great mass of invalid packets as for individual or a small number of packets.

Examples of such packets are:

- Mass-produced TCP packets with a set SYN flag to produce half-open TCP connections (SYN flooding attack).
- Packets with the same IP sender address and IP recipient address (Land attack).
- Mass-produced ICMP packets with the broadcast address of a network as target address (Smurf attack).
- Fragmented IP packets with overlapping offset fields (Teardrop attack).
- ICMP packets that are larger than the maximum permitted size (65,535 Bytes) of IPv4 packets (Ping-of-death attack).
- Uncorrelated reply packets (i.e. packets which cannot be correlated to any request).

Sometimes the relevant behaviour of the network device will be configured. In other cases, the behaviour of the network device may only be verified by the relevant tests.

Security Objective references: PROTECTED COMMUNICATIONS, HARDENING.

Test case: Refer to Test Case in clause 4.4.4.

4.2.6.2.3 GTP-C Filtering

Requirement Name: GTP-C Filtering

Requirement Description:

The following capability is conditionally required:

- For each message of a GTP-C-based protocol, it shall be possible to check whether the sender of this message is authorized to send a message pertaining to this protocol.

NOTE 1: The check could be performed e.g. against a whitelist or blacklist of permitted message type / sender identity combinations.

- At least the following actions should be supported when the check is satisfied:
    - Discard: the matching message is discarded.
    - Accept: the matching message is accepted.
    - Account: the matching message is accounted for, i.e. a counter for the rule is incremented. This action can be combined with the previous ones. This feature is useful to monitor traffic before its blocking.

This requirement is conditional in the following sense: It is required that at least one of the following two statements holds:

- The Network Product supports the capability described above and this is stated in the product documentation.
- The Network Product's product documentation states that the capability is not supported and that the Network Product needs to be deployed together with a separate entity which provides the capability described above.

NOTE 2: Such a separate entity could e.g. be a GTP Firewall.

NOTE 3: Test cases for this separate entity are not provided in the present document, but are believed to be similar to them.

NOTE 4: The test cases are only applicable to all network product classes utilizing GTP-C based protocol.

Security Objective references: tba.

Test case:

The test case described here apply only when GTP-C filtering is provided on the Network Product itself.

Test Name: TC_GTP-C_FILTERING

Purpose:

To verify that the network product provides filtering functionalities for incoming GTP-C messages. In particular this test case verifies that:

1. The network product provides filtering of incoming GTP-C messages on any interface.
2. It is possible to block all GTP-C messages on those network product interfaces where they are unwanted.
3. It is possible to specify defined actions for each rule.

Procedure and execution steps:

Pre-Conditions:

- The network product has at least two physical interfaces, named if1 and if2.
- The tester has the privileges to configure GTP-C filtering on the network product.
- The manufacturer declares that the GTP-C filtering is supported.
- The manufacturer includes a guideline to configure the GTP-C filtering in the documentation accompanying the network product.
- A network traffic generator or a pcap file containing the GTP-C messages is available.
- A network traffic analyser on the network product (e.g. tcpdump) is available.

Execution Steps

1. The tester log in the network product.

2. The tester configures the network product with the following rules:

    a) Accept only GTP-C EchoRequest messages on if1.

    b) Discard all GTP-C messages on if2.

    c) For each rule above the accounting is also enabled.

3. The tester turns on the network traffic analyser on if2.

4. The tester sends on if2 EchoRequest messages replying a pcap file or using a network generator.

    a) Using the network analyser the tester verifies that the network product correctly receives the EchoRequest messages on if2.

    b) Using the accounting, the tester verifies that the messages are discarded and that any response is sent back by the network product.

5. The tester sends to if1 EchoRequest messages replying a pcap file or using a network generator.

    a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

    b) The tester verifies that the GTP-C EchoRequest messages are not discarded because EchoResponse messages are sent back by the network product.

6. The tester verifies that the matching messages are correctly accounted for both rules.

7. The tester sends to if1 GTP-C messages different from EchoRequest replying a pcap file or using a network generator.

    a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

    b) Using the accounting, the tester verifies that the messages are discarded and that any response is sent back by the network product.

8. The tester deletes the previous rules and configures a new rule, i.e. to accept only GTP-C EchoRequest on if1 coming from a certain IP Address named IP1.

9. The tester sends GTP-C EchoRequest messages with source IP Address set to IP1:

    a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

    b) The tester verifies that the GTP-C EchoRequest messages are not discarded and EchoResponse messages are sent back by the network product.

10. The tester sends GTP-C EchoRequest messages with source IP Address set to IP2 different from IP1 using a network traffic generator or replying a pcap file.

    a) Using the network analyser the tester verifies that the messages are correctly received by the network product.

    b) The tester verifies that the GTP-C EchoRequest messages are discarded and that no EchoResponse messages are sent back.

Expected Results:

- For steps 4, 5, 6 and 7 the tester receives GTP-C EchoResponse messages from if1 only.

- For steps 4, 5, 6 and 7 the messages matching the rules are correctly accounted.

- For steps 8, 9, 10 the tester receives GTP-C EchoResponse messages only for the authorized source IP address.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The used tool(s) name and version information
- Settings and configurations used
- Pcap trace
- Screenshot

Test result (Passed or not)

4.2.6.2.4 GTP-U Filtering

Requirement Name: GTP-U Filtering

Requirement Description:

The following capability is conditionally required:

- For each message of a GTP-U-based protocol, it shall be possible to check whether the sender of this message is authorized to send a message pertaining to this protocol.

NOTE 1: The check could be performed e.g. against a whitelist or blacklist of permitted message type / sender identity combinations.

- At least the following actions should be supported when the check is satisfied:
  - Discard: the matching message is discarded.
  - Accept: the matching message is accepted.
  - Account: the matching message is accounted for, i.e. a counter for the rule is incremented. This action can be combined with the previous ones. This feature is useful to monitor traffic before its blocking.

This requirement is conditional in the following sense: It is required that at least one of the following two statements holds:

- The Network Product supports the capability described above and this is stated in the product documentation.
- The Network Product's product documentation states that the capability is not supported and that the Network Product needs to be deployed together with a separate entity which provides the capability described above.

NOTE 2: Such a separate entity could e.g. be a GTP Firewall.

NOTE 3: Test cases for this separate entity are not provided in the present document, but are believed to be similar to them.

NOTE 4: The test cases are only applicable to all network product classes utilizing GTP-U based protocol.

Security Objective references: tba.

Test case:

The test case described here apply only when GTP-U filtering is provided on the Network Product itself.

Test Name: TC_GTP-U_FILTERING

Purpose:

To verify that the network product provides filtering functionalities for incoming GTP-U messages. In particular this test case verifies that:

1. The network product provides filtering of incoming GTP-U messages on any interface.

2. It is possible to block all GTP-U messages on those network product interfaces where they are unwanted.

3. It is possible to specify defined actions for each rule.

Procedure and execution steps:

Pre-Conditions:

- The network product has at leastone physical interface named if1 and may have another physical interface named if2 .

- The tester has the privileges to configure GTP-U filtering on the network product.

- The manufacturer declares that the GTP-U filtering is supported.

- The manufacturer includes a guideline to configure the GTP-U filtering in the documentation accompanying the network product.

- A network traffic generator or a pcap file containing the GTP-U messages is available.

- A network traffic analyser on the network product (e.g. tcpdump) is available.

NOTE: If the network product has only one physical interface named if1, execution steps on if2 are not needed.

Execution Steps

1. The tester log in the network product.

2. The tester configures the network product with the following rules:

    a) Accept only GTP-U EchoRequest messages on if1.

    b) Discard all GTP-U messages on if2.

    c) For each rule above the accounting is also enabled.

3. The tester turns on the network traffic analyser on if2.

4. The tester sends on if2 EchoRequest messages replying a pcap file or using a network generator.

    a) Using the network analyser the tester verifies that the network product correctly receives the EchoRequest messages on if2.

    b) Using the accounting, the tester verifies that the messages are discarded and that any response is sent back by the network product.

5. The tester sends to if1 EchoRequest messages replying a pcap file or using a network generator.

    a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

    b) The tester verifies that the GTP-U EchoRequest messages are not discarded because EchoResponse messages are sent back by the network product.

6. The tester verifies that the matching messages are correctly accounted for both rules.

7. The tester sends to if1 GTP-U messages different from EchoRequest replying a pcap file or using a network generator.

    a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

    b) Using the accounting, the tester verifies that the messages are discarded and that any response is sent back by the network product.

8. The tester deletes the previous rules and configures a new rule, i.e. to accept only GTP-U EchoRequest on if1 coming from a certain IP Address named IP1.

9. The tester sends GTP-U EchoRequest messages with source IP Address set to IP1:

   a) Using the network analyser, the tester verifies that the messages are correctly received by the network product.

   b) The tester verifies that the GTP-U EchoRequest messages are not discarded and EchoResponse messages are sent back by the network product.

10. The tester sends GTP-U EchoRequest messages with source IP Address set to IP2 different from IP1 using a network traffic generator or replying a pcap file.

    a) Using the network analyser the tester verifies that the messages are correctly received by the network product.

    b) The tester verifies that the GTP-U EchoRequest messages are discarded and that no EchoResponse messages are sent back.

Expected Results:

- For steps 4, 5, 6 and 7 the tester receives GTP-U EchoResponse messages from if1 only.

- For steps 4, 5, 6 and 7 the messages matching the rules are correctly accounted.

- For steps 8, 9, 10 the tester receives GTP-U EchoResponse messages only for the authorized source IP address.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The used tool(s) name and version information

- Settings and configurations used

- Pcap trace

- Screenshot

Test result (Passed or not)

4.3 Security requirements and related test cases related to hardening

4.3.1 Introduction

The requirements proposed hereafter (with the relative test cases) aim to securing network products (including the network functions in service-based architecture) by reducing its surface of vulnerability. In particular the identified requirements aim to ensure that all the default network product configurations (including operating system software, firmware and applications) are appropriately set.

4.3.2 Technical Baseline

4.3.2.1 No unnecessary or insecure services / protocols

Requirement Name: No unnecessary or insecure services / protocols

Requirement Description:

The network product shall only run protocol handlers and services which are needed for its operation, and which do not have any known security vulnerabilities. In particular, by default the following services shall be initially configured to be disabled on the network product by the vendor except if services are needed during deployment. In that case those services shall be disabled according to vendor's instructions after deployment is done. Disabled protocols may still need to be enabled for other reasons by the operators, e.g. remote diagnostics.

- FTP
- TFTP
- Telnet
- rlogin, RCP, RSH
- HTTP
- SNMPv1 and v2
- SSHv1
- TCP/UDP Small Servers (Echo, Chargen, Discard and Daytime)
- Finger
- BOOTP server
- Discovery protocols (CDP, LLDP)
- IP Identification Service (Identd)
- PAD
- MOP

NOTE 1: As an alternative to disabling the HTTP service, it is also possible for this service to remain active for reasons of user friendliness. In this case, however, queries to the web service may not be answered directly on this port but from a redirected to HTTPS service.

NOTE 2: Full documentation of required protocols and services of the network product and their purpose needs to be provided by the vendor as prerequisite for the test case.

Test Case: TBA

Test Name: TC_NO_UNNECESSARY_SERVICE

Purpose:

To ensure that on all network interfaces, there are no unsecure services or protocols that might be running.

Procedure and execution steps:

Pre-Conditions:

A list of all required network protocols and services containing at least the following information shall be included in the documentation accompanying the Network Product:

- protocol handlers and services needed for the operation of network product;
- their open ports and associated services;
- and a description of their purposes.

The tool used shall be capable to detect and identify the protocol handlers and running services in the system.

Execution Steps

The accredited evaluator's test lab is required to execute the following steps:

1. Verification of the compliance to the prerequisites:
    a. Verification that the list of available network services and protocol handlers is available in the documentation of the Network Product.
    b. Validation that all entries in the list are meaningful and reasonably necessary for the operation of the Network Product class.
2. Identification of the network services and protocol handlers by means of capable tools or any other suitable testing means.
3. Validation that there are no entries in the list of network services and handlers apart from the ones that have been mentioned and deemed necessary for the operation of the Network Product in the attached documentation.
4. The tester shall reboot the network product and re-execute execution steps 2 and 3 without further configuration.

Expected Results:

The report will contain:

- The names and version of the tool(s) used.
- Information of all the protocol handlers and services running in the network product.

Result will show:

- There are no unnecessary services running in the network product except for the ones which are deemed necessary for its operation.
- Any undocumented services running on the network product should be highlighted and brought out in the report.
- The network product behaves the same after reboot as before.

Expected format of evidence:

A report provided by the testing agency which will consist of the following information:

- The used tool(s) name and version information
- Settings and configurations used
- The output pertaining to the test case performed and
- The test results i.e. services existing or not existing in the MME

4.3.2.2 Restricted reachability of services

Requirement Name: The network product shall restrict the reachability of services

Requirement Description:

The network product shall restrict the reachability of services so that they can only be reached on interfaces where their usage is required. On interfaces were services are active, the reachability should be limited to legitimate communication peers. This limitation shall be realized on the network product itself (without measures (e.g. firewall) at network side) according to the requirement detailed in clause 4.2.6.2.1 Packet Filtering.

EXAMPLE: Administrative services (e.g. SSH, HTTPS, RDP) shall be restricted to interfaces in the management network to support separation of management traffic from user traffic.

Test Case:

Test Name*:* TC_RESTRICTED_REACHIBILITY_OF_SERVICES

Purpose:

To verify that it is possible to bind the services only to the interfaces from which they are expected to be reachable.

NOTE: The test case developed for the requirement " 4.2.6.2.1 Packet Filtering" implicitly verifies that the network product permits to limit the reachability of the services only to legitimate communication peers,

Procedure and execution steps:

Pre-Conditions:

- The vendor shall declare, in the documentation accompanying the network product if the network product supports the capability to restrict services reachability to only the nodes authorized to access them. In this case, the vendor shall detail how this capability can be configured.
- A list of all required network protocols and services containing at least the following information shall be included in the documentation accompanying the Network Product:
  - protocol handlers and services needed for the operation of network product;
  - their open ports and associated services;
  - the configuration options;
  - and a description of their purposes.
- The network product is configured such that the required network protocols and services (as described in the network product documentation) are setup and each service is bound to an IP address of a specific network interface (e.g. IP1 which is the ip address of if1). Configuration may occur automatically during the initialization phase of the network product or manually as defined in the network product administration documentation.
- The network product shall have at least two interfaces enabled, if1 and if2 respectively configured with IP Address IP1 and IP2.
- The tester has administrative privileges.
- A tester machine equipped with a network port scanner tool is available.

Execution Steps

1. The tester runs a network port scanner (e.g. nmap) towards if1 and verifies that the configured services are open/reachable.
2. The tester runs a network port scanner (e.g. nmap) towards if2 and verifies that the configured services are not open/reachable.

Expected Results:

Services can be enabled on per-interface basis.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The network product configuration
- Pcap files
- Screenshot
- Network port scanner results (e.g. files containing this results)
- Test result (Passed or not)

4.3.2.3 No unused software

Requirement Name: Unused software shall not be installed or shall be uninstalled

Requirement Description:

Unused software components or parts of software which are not needed for operation or functionality of the network product shall not be installed or shall be deleted after installation. This includes also parts of a software, which will be installed as examples but typically not be used (e.g. default web pages, example databases, test data).

Test Case:

Test Name: TC_NO_UNUSED_SOFTWARE

Purpose:

To ensure that there is no unused software or associated components that might be installed in the network product which are not required for its operation or functionality.

Procedure and execution steps:

Pre-Conditions:

A list of all available software and libraries and associated components containing at least the following information shall be included in the documentation accompanying the Network Product:

- name of the software / library;
- version of the software / library installed;
- list of dependencies and versions;
- any add-ons and functions;
- any special hardware/debugging ports;
- software support type;
- licensing information;
- brief description of their purpose.

Execution Steps

The accredited evaluator's test lab is required to execute the following steps:

1. Verification of the compliance to the prerequisites:
    a. Verification that the list of software / libraries and components is available in the documentation of the Network Product.
    b. Validation that all entries in the list of software / libraries and components are meaningful and reasonably necessary for the operation of the Network Product class.
2. Identification of the software / libraries or components which are installed in the system using any suitable command line tools or any other suitable means of determination.
3. Validation that there are no entries in the list of software / libraries installed in the system apart from the ones that have been mentioned and deemed necessary for the operation of the network product in the attached documentation.
4. Based on his/her experience, the tester will check for known default example files for software installed on the system.

Expected Results:

The report will contain the names and version of the tool(s) used for finding out what software /libraries is installed in the system. The detailed report will contain the name and version information of all the software / libraries installed in the system generated by the tool.

The list of all available software / libraries which has been deemed necessary for the operation of the network product by the vendor shall also be included as the test result. Any software / library not in the list of allowed software / libraries will be highlighted and brought out as a part of the report.

There should be no unnecessary software / library installed in the network product except for the ones which are deemed necessary for its operation.

There should be no more default example files for the installed software on the system.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The used tool(s) name and version information,
- Settings and configurations used
- the output pertaining to the test case performed and,
- the test results i.e. list of allowed and disallowed software

4.3.2.4 No unused functions

Requirement Name: Unused functions of the network products' software and hardware shall be deactivated.

Requirement Description:

During installation of software and hardware often functions will be activated that are not required for operation or function of the system. If unused functions of software cannot be deleted or deinstalled individually as required in clause "5.3.2.3 No unused software" of the present document, such functions shall be deactivated in the configuration of the network product permanently.

Also hardware functions which are not required for operation or function of the system (e.g. unused interfaces) shall be permanently deactivated. Permanently means that they shall not be reactivated again after network product reboot.

EXAMPLE: A debugging function in software which can be used for troubleshooting shall not be activated during normal operation of the network product.

Test Case:

Test Name: TC_NO_UNUSED_FUNCTIONS

Purpose:

To ensure that there is no unused hardware or software functions that are not deactivated in the network product which are not required for its operation or functionality.

Procedure and execution steps:

Pre-Conditions:

A list of all available software and associated components containing at least the following information shall be included in the documentation accompanying the Network Product:

- name of the software;
- version of the software installed;
- list of dependencies and versions;
- any add-ons and functions;
- any special hardware/debugging ports;

- software support type;
- licensing information;
- requirement during functioning of system;
- brief description of their purpose.

Execution Steps:

The accredited evaluator's test lab is required to execute the following steps:

1. Identification of the hardware and software functions which are installed in the system or might have been disabled using any suitable command line tools or any other suitable means of determination.
2. Validate that there are no entries in the list of hardware and software functions installed in the system apart from the ones that have been mentioned and deemed necessary for the operation of the network product in the attached documentation.

Expected Results:

The report will contain the names and version of the tool(s) used for finding out what software and associated function is installed in the system. The detailed report will contain the name and version information of all the software and components installed in the system generated by the test tool.

The list of all available software which has been deemed necessary for the operation of the network product by the vendor shall also be included as the test result. Any software not in the list of allowed software will be highlighted and brought out as a part of the report.

There should be no unused function that is not deactivated in the network product except for the ones which are deemed necessary for its operation.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The used tool(s) name and version information
- Settings and configurations used
- The list of software and associated functions
- the test results i.e. allowed list of functions

4.3.2.5 No unsupported components

Requirement Name: The network product shall not contain software and hardware components that are no longer supported by their vendor, producer or developer.

Requirement Description:

The network product shall not contain software and hardware components that are no longer supported by their vendor, producer or developer, such as components that have reached end-of-life or end-of-support. Excluded are components that have a special support contract. This contract shall guarantee the correction of vulnerabilities over components' lifetime.

Test Case:

Test Name: TC_NO_UNSUPPORTED_COMPONENTS

Purpose:

To ensure that there is no unsupported software that is running in the network product which is not supported anymore and has reached its end-of-life or end-of-support.

Procedure and execution steps:

Pre-Conditions:

A list of all available software and associated components containing at least the following information shall be included in the documentation accompanying the Network Product:

- name of the software;
- version of the software installed;
- list of dependencies and versions;
- any add-ons and functions;
- any special hardware/debugging ports;
- software support type;
- licensing information;
- requirement during functioning of system;
- brief description of their purpose.

Execution Steps

The accredited evaluator's test lab is required to execute the following steps:

1. Identification of the hardware and software components, version information and the kind of support available for the software provided by the vendor, the producer, the developer or other contractual partner of the operator using any tool or any other suitable means of determination.
2. Validate that there are no entries in the list of hardware and software installed in the system which are not supported as given by the vendor of network product in the attached documentation.

Expected Results:

The report will contain the names and versions of the tool(s) used for finding out what software and hardware components are installed in the system. The detailed report will contain the name and version of the software and hardware used in the system, and the period of support for each of these components.

The list of all available software and hardware components and their associated support information which has been deemed necessary for the operation of the network product by the vendor shall also be included as the test result. Any software or component which is not supported any longer by the vendor will be highlighted and brought out as a part of the report.

There should be no software installed in the network product which is unsupported as of the day of testing.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The used tool(s) name and version information
- Software and hardware components used in the network product
- the test results i.e. support information of each listing

4.3.2.6 Remote login restrictions for privileged users

Requirement Name: Remote login restrictions for privileged users

Requirement Description: Direct login as root or equivalent highest privileged user shall be limited to the system console only. Root user will not be allowed to login to the system remotely.

Test Case:

Test Name: TC_REMOTE_LOGIN_RESTRICTIONS_PRIVILEGED_USERS

Purpose:

Verify that root or equivalent highest privileged user will not be allowed to login to the system remotely.

Procedure and execution steps:

Pre-Condition:

A document that describes the interfaces to the network product and how the tester can login to them remotely.

Execution Steps

Execute the following steps:

1. The tester tries to remotely login to the network product using the credentials of the root or equivalent highest privileged user via the interfaces as described in the documentation.
2. The tester tries to login to the network product using the credentials of the root or equivalent highest privileged user from the physical console of the system.

Expected Results:

The tester is not able to login to the system remotely using the root credentials.

The tester is able to login to the system from the physical console using the root credentials.

Expected format of evidence:

A Pass/Fail result.

4.3.2.7 Filesystem Authorization privileges

Requirement Name: Filesystem Authorization privileges.

Requirement Description: The system shall be designed to ensure that only users that are authorized to modify files, data, directories or file systems have the necessary privileges to do so.

EXAMPLE: On unix® systems a 'sticky' bit may be set on all directories where all users have write permissions. This ensures that only the file's owner, the directory's owner, or root user can rename or delete the file. Without the sticky bit being set, any user that has write and execute permissions for the directory can rename or delete files within the directory, regardless of the file's owner.

Test Case:

Test Name: TC_FILESYSTEM_AUTHORIZATION_PRIVILEGES

Purpose:

Verify that only users that are authorized to modify files, data, directories or file systems have the necessary privileges to do so.

Procedure and execution steps:

Pre-Condition:

A document describing how access control is configured for the filesystems in the network product shall be provided by the vendor.

Execution Steps

Execute the following steps:

1. The tester checks that OS-level permissions are configured correctly for users that are authorized to modify files, data, directories or file systems on the system.

2. The tester tries to modify the files and directories for which the user has the necessary privileges.

3. The tester tries to modify the files and directories for which the user doesn't have the necessary privileges.

Expected Results:

The OS-level access permissions are set correctly for the users.

The users can only modify files, data, directories or file systems for which he has the necessary privileges to do so.

Expected format of evidence:

Screenshot containing the configuration file showing the OS-level permissions are set correctly.

4.3.3 Operating Systems

4.3.3.1 General operating system requirements and test cases

4.3.3.1.1 IP-Source address spoofing mitigation

Requirement Name: IP-Source address spoofing mitigation

Requirement Description:

Systems shall not process IP packets if their source address is not reachable via the incoming interface. Implementation example: Use of "Reverse Path Filter" (RPF) provides this function.

Test Case:

The test for this requirement can be carried out using a suitable tool or manually by performing the steps described below. If a tool is used then the tester needs to provide evidence, e.g. by referring to the documentation of the tool, that the tool actually provides functionality equivalent to the steps described below.

Test Name: TC_IP_SPOOFING_MITIGATION

Purpose:

To verify that the network product provides anti-spoofing function that is, before a packet is processed, the network product checks whether the source IP of the received packet is reachable through the interface it comes in.

To verify that if the received packet source address is not routable through the interface on which it comes, then the network product drops this packet.

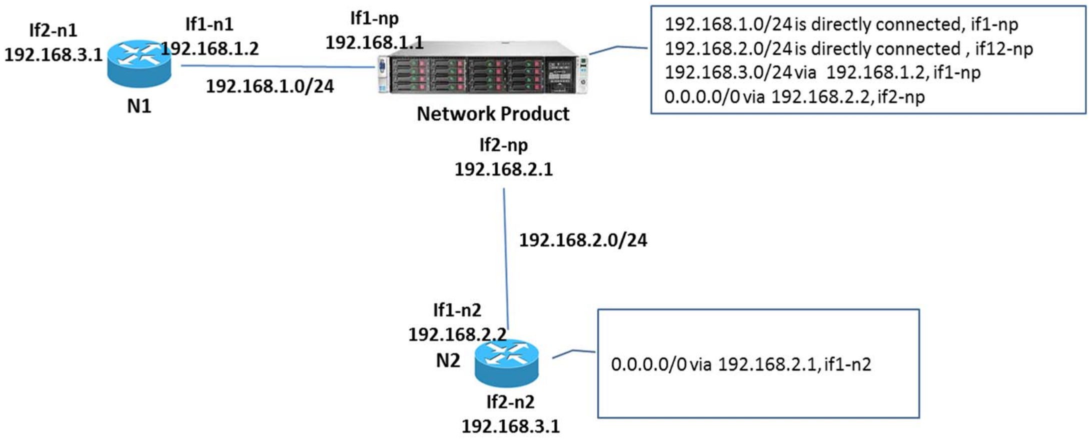
Procedure and execution steps:

Pre-Conditions:

- A node N1 is available with:
  - Two interfaces named respectively if1-n1 connected to the network product and if2-n1 to which the tester connects a tester machine

- routing capabilities
- if2-n1 has a static IP address (e.g. 192.168.3.1 belonging to the subnet 192.168.3.0/24)

- A node N2 is available with:
  - Two interfaces named respectively if1-n2 connected to the network product and if2-n2 to which the tester connects a tester machine
  - Routing capabilities. In particular N2 has a default route to if1-np subnet via if2-np (e.g. 192.168.2.1)
  - if2-n2 has a static IP address . This ip is the same as if2-n1 (e.g. 192.168.3.1 belonging to the subnet 192.168.3.0/24)
- The network product has at least 2 enabled interfaces said if1-np and if2-np:
  - The interface if1-np is connected to interface if1-n1 of the node N1 on the subnet, e.g., 192.168.1.0/24.
  - The interface if2-np is connected to interface if1-n2 of the node N2 on the subnet, e.g., 192.168.2.0/24.
  - The network product is configured with a static route for the subnet where if2-n1 is connected to (e.g. 192.168.3.0/24), so this subnet can be reached via if1-n1 through if1-np.



192.168.1.0/24 is directly connected, if1-np
192.168.2.0/24 is directly connected , if12-np
192.168.3.0/24 via 192.168.1.2, if1-np
0.0.0.0/0 via 192.168.2.2, if2-np

0.0.0.0/0 via 192.168.2.1, if1-n2

Figure 1: Configurations for the network product, N1 and N2

- The vendor shall declare, in the documentation accompanying the network product, the supported anti-spoofing mechanism (e.g. RPF or similar function) and if it is enabled for all interfaces (e.g. net.ipv4.conf.all.rp_filter = 1 and net.ipv4.conf.default.rp_filter = 1 in the linux sysctl.conf file) or per interface bases.
- The vendor shall declare if the dropped packets can be logged and how to enable this logging
- The tester has administrator privileges
- A tester machine is available and configured with:
  - A static IP address belonging to the subnet where if2-n1 and if2-n2 are connected to (e.g. 192.168.3.2/24)
  - A default gateway set to if2-n1 and if2-n2 IP Address (e.g. 192.168.3.1)
- A network traffic analyser (e.g. tcpdump) on the network product is available

Execution Steps

1. The tester starts to send ping messages to if1-np interface of the network product.
2. The tester verifies, through the network traffic analyser, that the ping reaches correctly the if1-np interface and that responses are sent back.

3. The tester disconnects the tester machine from if2-n1 interface of the node N1 and reconnects it to the interface if2-n2 of the node N2:

    - The testers uses the same network configuration of the tester machine.

    - The tester sends ping messages to if1-np interface of the network product.

    - The tester verifies, through the network traffic analyser, that the pings reach the if1-np interface of the network product, but they are dropped and no response is sent back since the source of the received packet is not reachable through the interface it came in.

    - The tester sends ping messages to if2-np interface of the network product.

    - The tester verifies, through the network traffic analyser, that the pings reach the if2-np interface of the network product, but they are dropped and no response is sent back since there is a default route via if2-np.

    - If the dropped packets are logged, the testers verifies that these packets are recorded.

Expected Results:

The network product supports an anti-spoofing mechanism (e.g. the RPF function) and it accepts a packet only if it reaches the network product on the expected interface (i.e. this packet has a source ip address belonging to the same network as the interface where it came in or if it is routable through the interface on which it came in), otherwise it discards the packet.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The user settings and configurations

- Pcap files

- Log file if available

- Test result (Passed or not)

4.3.3.1.2 Minimized kernel network functions

Requirement Name: Minimized kernel network functions.

Requirement Description:

Kernel based network functions not needed for the operation of the network element shall be deactivated.

In particular the following ones shall be disabled by default:

- IP Packet Forwarding between different interfaces of the network product.

NOTE: The above text does not preclude that IP Packet Forwarding can be enabled in certain deployment scenarios.

- Proxy ARP (to prevent resource exhaustion attack and man-in-the-middle attacks.

- Directed broadcast (to prevent Smurf, Denial of Service attack and others like it.

- IPv4 Multicast handling. In particular all packets with IP source or destination address belonging to the multicast IP ranges (224.0.0.0 through 239.255.255.255) shall be discarded by default and multicast route caching and forwarding shall be disabled to prevent smurf and fraggle attacks. A configuration option shall be available to enable the IPv4 multicast handling if required.

- Gratuitous ARP messages (to prevent ARP Cache Poisoning attacks [ef]). A Gratuitous ARP request can be used mainly to inform the neighbours about the change in the MAC for the specified IP and consequently to update their ARP tables or to update the switches with the new MAC address or to defend link-local IP addresses in the

Zeroconf protocol. By default, the network product shall not send Unsolicited ARP and any incoming Gratuitous ARP requests shall be discarded.

NOTE: The above text does not preclude that Gratuitous ARP can be enabled in certain deployment scenarios.

Answering routine for broadcast ICMP packets. In particular all ICMP ECHO and TIMESTAMP requests sent to network product via broadcast/multicast shall not be answered by default.

Test Case:

Test Name: TC_IP_FWD_DISABLING

Purpose:

Verify that IP Packet Forwarding is disabled by default on the network product. In particular this test case verifies that a packet received by a network product interface but directed to a host on a different network is not routed by the network product

Procedure and execution steps:

Pre-Conditions:

- The network product has at least 2 different physical or logical Ethernet interfaces.
- Host 1 is connected to Interface 1 on subnet A and Host 2 is connected to Interface 2 on subnet B.
- Network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product is available.

Execution Steps

- If the feature is available in a configuration file, verify that it is disabled by default.
- Send a packet from Host 1 on subnet A to Host 2 on subnet B with the network product configured as a default gateway.
- Verify that the packet is correctly received by the network product (logged by the network traffic analyser) but it is not routed to Host 2.

Expected Results:

The packet is not routed by the network product and Host 2 does not receive it.

Expected format of evidence:

Pcap trace of the received packet

Test Name: TC_PROXY_ARP_DISABLING

Purpose:

Verify that the Proxy ARP feature is disabled by default on the network product. In particular this test case verifies that the network product does not respond to ARP requests intended for another host.

Procedure and execution steps:

Pre-Conditions:

- The network product shall have at least 2 different physical or logical Ethernet interface IF1 and IF2. E.g.

  - Host 1 is connected to IF1 on subnet A (for example 172.16.10.0/16).
  - Host 2 is connected to IF2 on subnet B (for example 172.16.20.0/24).
- Network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product is available.

Execution Steps

1. If the feature is available in a configuration file, verify that it is disabled by default.
2. Broadcast an ARP request from Host 1 on Subnet A to discover the MAC of Host 2 on subnet B. Since the ARP request is a broadcast, it reaches all nodes in the Subnet A, which include the IF1 interface of the network product, but it does not reach Host 2.
3. Verify that the network product correctly receives this packet but that it does not send an ARP reply to Host 1 with its own MAC address.

Expected Results:

No Arp Reply is received by Host 1.

Expected format of evidence:

Pcap trace, snapshot of ARP Cache of Host 1

Test Name: TC_DIRECTED_BROAD_DISABLING

Purpose:

Verify that the Directed broadcast is disabled by default on the network product. In particular this test case verifies that a packet received by a network product whose destination address is a valid broadcast address is dropped.

Procedure and execution steps:

Pre-Conditions:

- The network product has at least 2 different physical or logical Ethernet interface IF1 and IF2.
- Host 1 is connected to IF1 on Subnet A and Host 2 is connected to IF2 on Subnet B.
- Network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product is available.

Execution Steps

1. If the feature is available in a configuration file, verify that it is disabled by default.
2. Send an IP packet from Host 1 whose IP destination address is a valid broadcast address belonging to the subnet B.
3. Verify that the Host 2 on Subnet B does not receive the packet because it will be dropped by the network product, rather than being broadcasted.

Expected Results:

The packet is not broadcasted by the network product and Host 2 cannot receive it.

Expected format of evidence:

Pcap trace showing that packet from host 1only incomes to the network product.

Test Name: TC_ IP_MULTICAST_HANDLING

Purpose:

Verify that IP Multicast is disabled by default on the network product. In particular this test case verifies that packets with IP source or destination address belonging to the multicast IP ranges (224.0.0.0 through 239.255.255.255) are not handled by the network product.

Procedure and execution steps:

Pre-Conditions:

- Network traffic analyser on the network product or an external traffic analyser directly connected to the network product is available.

Execution Steps

1. If the feature is available in a configuration file, verify that it is disabled by default.
2. Verify that none of the network product's interfaces is running Multicast (e.g. typing command *ip maddr* or *ifconfig* on any Unix® based platform)

Expected Results:

No interface is running multicast protocols

Expected format of evidence:

Screenshot containing command output.

Test Name: TC_GRATUITOUS_ARP_DISABLING

Purpose:

Verify that the Gratuitous ARP feature is disabled by default on the network product. In particular this test case verifies that the network product cannot send gratuitous ARP requests and that the network product discards incoming Gratuitous ARP requests.

Procedure and execution steps:

Pre-Conditions:

- Network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product is available.
- Host 1 is connected to the network product
- The network product ARP Cache already contains an entry for Host 1

- The network product documentation does not state any reason why gratuitous ARP may be deliberately enabled in order to satisfy certain deployment scenarios. If the network product documentation does, however, state that the usage of gratuitous ARP is enabled in certain deployment scenarios, then this test case is not applicable (refer to the NOTE in the requirement).

Execution Steps

1. If the feature is available in a configuration file, verify that it is disabled by default.
2. Send a Gratuitous ARP request from Host 1, i.e. an ARP *request* where the source and destination IP are both set to an IP address different from the one already cached in the network product ARP Cache for Host 1 and the destination MAC is the broadcast address ff:ff:ff:ff:ff:ff.
3. Verify that the network product correctly receives this packet but discards it and that the ARP Cache is not updated.
4. Send a Gratuitous ARP request i.e. an ARP reply where the source and destination IP are both set to an IP address different from the one already cached in the network product ARP Cache for Host 1 and the destination MAC is the broadcast address ff:ff:ff:ff:ff:ff.
5. Verify that the network product correctly receives this packet but discards it and that the ARP Cache is not updated.

Expected Results:

The network product ARP Cache is not updated**.**

Expected format of evidence:

Snapshot of the network product ARP Cache

Test Name: TC_BROADCAST_ICMP_HANDLING

Purpose:

Verify that responses to ICMP broadcast packets are disabled by default on the network product . In particular this test case verifies that all ICMP ECHO and TIMESTAMP requests sent to the network product via broadcast/multicast are not answered.

Procedure and execution steps:

Pre-Conditions:

- The network product has at least one physical or logical Ethernet interface IF1 connected to a host, Host 1.
- Network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product is available.

Execution Steps

1. If the feature is available in a configuration file, verify that it is disabled by default. .
2. Send an ICMP ECHO message from Host 1 to ping a broadcast address (such as 255.255.255.255, or 192.168.1.255 on a 192.168.1.0/24 subnet)

   3. Verify that the network product doesn't respond to the ping.

4. Send an ICMP timestamp request (ICMP type 13) from host 1 to a broadcast address (such as 255.255.255.255, or 192.168.1.255 on a 192.168.1.0/24 subnet).

5. Verify that the network product doesn't respond to the timestamp request.

Expected Results:

The network product doesn't respond to any ICMP packet with a broadcast address.

Expected format of evidence:

Pcap trace showing that the ICMP ECHO/ ICMP timestamp packets are received by the network product but no responses are generated by the network product.

4.3.3.1.3 No automatic launch of removable media

Requirement Name: No automatic launch of removable media

Requirement Description:

The network product shall not automatically launch any application when removable media device such as CD-, DVD-, USB-Sticks or USB-Storage drive is connected. If the operating system supports an automatic launch, it shall be deactivated unless it is required to support availability requirements.

Test Case:

Test Name: TC_NO_AUTO_LAUNCH_OF_REMOVABLE_MEDIA

Purpose:

To verify that the network product does not launch any applications automatically when a removable media device is connected. Any such feature should be deactivated.

Procedure and execution steps:

Pre-Condition

If the network product is provisioned with the necessary physical ports/drives (CD/DVD drive, USB port, etc.) then the test case applies.

Execution Steps

1. The tester log in the network product.
2. The tester inserts a removable media device (CD-, DVD-, USB-Sticks and/or USB-Storage drives) in the network product.

Expected Results:

The network product does not launch any applications to open the contents in the removable media device.

In Linux® machines, the removable media device is not automatically mounted in the filesystem.

Expected format of evidence:

Evidence can be presented in the form of screenshot/screen-capture on how the network product responds when any removable media device is attached to it.

4.3.3.1.4 SYN Flood Prevention

Requirement Name: Syn Flood Prevention

Requirement Description:

The network product shall support a mechanism to prevent Syn Flood attacks (e.g. implement the TCP Syn Cookie technique in the TCP stack by setting net.ipv4.tcp_syncookies = 1 in the linux sysctl.conf file). This feature shall be enabled by default.

Test Case:

Test Name: TC_SYN_FLOOD_PREVENTION

Purpose:

Verify that the Network Product supports a Syn Flood Prevention technique.

Procedure and execution steps:

Pre-Conditions:

- The Network Product is listening on a TCP port one of its interfaces.
- A network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product is available.
- A host is connected to the Network Product interface and it is equipped with a tool able to reproduce a Syn Flood attack (e.g. nmap or hping)

Execution Steps

1. The tester configures the tool to send a huge amount of TCP Syn packets against the Network Product (e.g. `hping3 -i` <waiting time between each packet> `-S -p` <TCP port> `-c` <Number of packets> <MME IP>)
2. The Network Product is still up and running normally, its services are still available and reachable, the memory is not exhausted, there is no crash.

Expected Results:

The Network Product does not become inoperative.

Expected format of evidence:

A Pass/Fail result provided by the tester.

4.3.3.1.5 Protection from buffer overflows

Requirement Name: Protection mechanisms against buffer overflows

Requirement Description:

The system shall support mechanisms for buffer overflow protection. Documentation which describes these buffer overflow mechanisms and also how to check that they have been enabled and/or implemented shall be provided.

NOTE: Void

NOTE: Void

Test Case:

Test Name: TC_PROTECTION_FROM_BUFFER_OVERFLOW

Purpose:

To ensure that the system supports mechanisms that protect against buffer overflow.

Procedure and execution steps:

Pre-Conditions:

1. A document which provides a detailed technical description of the system's buffer overflow protection mechanisms.

   If a standard buffer overflow mechanism from a 3rd party vendor is used then a reference to the standard feature in the 3rd party vendors documentation should be provided.

2. Test results from a test execution phase of buffer overflow protection mechanism testing.

Execution Steps:

The accredited evaluator's test lab is required to execute the following steps:

1. The tester verifies that there is:

   a) A technical description of the buffer overflow protection mechanisms that have been implemented on the system.

   b) Details of whether the buffer overflow protection mechanisms are implemented by default or if additional actions (e.g. scripts or commands manually executed) are required.

   c) If manually executed actions are required then detailed instructions should be included in the technical description.

2. The tester verifies that the test results:

   a) Describe test procedures used to verify the buffer overflow protection mechanisms,

   b) Contain data which demonstrates/indicates that the buffer overflow protection mechanisms described in the technical description document have been implemented.

   c) Contains details of the test set-up for the testing of the buffer overflow protection mechanisms. Where simulators and/or scripts are used to artificially create the conditions to trigger the buffer overflow protection mechanism then details of these should also be included.

Expected Results:

1. A technical description of the buffer overflow protection mechanisms that have been implemented on the system.

   - Details of whether the buffer overflow protection mechanisms are implemented by default or if additional actions (e.g. scripts or commands manually executed) are required.

   - If manually executed actions are required then detailed instructions should be included in the technical description.

2. The test results should:

   - Describe test procedures used to verify the buffer overflow protection mechanisms,

   - Contain data which demonstrates/indicates that the buffer overflow protection mechanisms described in the technical description document have been implemented.

   - Contain details of the test set-up for the testing of the buffer overflow protection mechanisms. Where simulators and/or scripts are used to artificially create the conditions to trigger the buffer overflow protection mechanism then details of these should also be included.

Expected format of evidence:

Documentation showing each of the points in the results sections.

4.3.3.1.6 External file system mount restrictions

Requirement Name: External file system mount restrictions

Requirement Description:

If normal users are allowed to mount external file systems (attached locally or via the network), OS-level restrictions shall be set properly in order to prevent privilege escalation or extended access permissions due to the contents of the mounted file systems.

Implementation example: In Linux® systems, administrators shall set the options nodev and nosuid in the /etc/fstab for all filesystems, which also have the "user" option.

NOTE: This requirement does not apply when the docker is used to mount file system.

Test Case:

Test Name: TC_EXTERNAL_FILE_SYSTEM_MOUNT_RESTRICTIONS

Purpose:

Verify that OS-level restrictions are set properly for users that are allowed to mount external file systems (attached locally or via the network). This is to prevent privilege escalation or extended access permissions due to the contents of the mounted file systems.

Procedure and execution steps:

Pre-Condition:

Tester has admin access to check and configure the external filesystem mount permissions in the OS.

Tester has username and password of a user in the network product that has external filesystem mount privileges.

Execution Steps

Execute the following steps:

1. The tester shall verify that OS-level restrictions are set properly in order to prevent privilege escalation due to the contents of the mounted file systems (e.g. In Linux® systems, administrators shall set the options nodev and nosuid in the /etc/fstab for all filesystems, which also have the "user" option). The tester checks that OS-level parameters are configured correctly on the system.

2. The tester mounts an external filesystem prepared by the tester with files exploiting privilege escalation methods (e.g. with writable SUID/GUID files).

3. The tester tries to gain privileged access to system by using a suitable privilege escalation method using the contents of the mounted file system and then confirms that privilege escalation doesn't happen.

Expected Results:

The OS-level restrictions are set properly in order to prevent privilege escalation or extended access permissions due to the contents of the mounted file systems.

Any privilege escalation method used by the tester should be blocked.

Expected format of evidence:

Screenshot containing the configuration file showing that OS-level restrictions are set properly for users that are allowed to mount external file systems.

4.3.4 Web Servers

4.3.4.1 General

Hardening requirements for Web servers of this section are well covered also by external sources, such as Center for Internet Security (CIS) benchmarks <https://benchmarks.cisecurity.org/index.cfm>. It is highly recommended to consult e.g. CIS, for the purpose of using automatic testing tools, for product-specific considerations, and for manual auditing, when testing the below listed requirements. If and when such mapping of requirements is used, i.e. to those of an external source, it needs to be well verified and documented that they cover the requirements of this section.

4.3.4.2 No system privileges for web server

Requirement Name: No system privileges for web server.

Requirement Description:

No web server processes shall run with system privileges. This is best achieved if the web server runs under an account that has minimum privileges. If a process is started by a user with system privileges, execution shall be transferred to a different user without system privileges after the start.

Test Case:

***Test Name*:** TC_NO_SYSTEM_PRIVILEGES_WEB_SERVER

Purpose:

Verify that the Web server is not run under system privileges.

Procedure and execution steps:

Pre-Conditions:

- The tester has needed administrative privileges.
- A tester machine is available.
- Recommended: an automatic assessment tool has been configured /script adapted in line with the Requirement Description.

Execution Steps

1. Check that no web server processes runs with system privileges. Check that this is the case even for processes that may have been started by a user with system privileges.
2. Check that relevant system settings and configurations are correct to ensure fulfilment of the requirement.

Expected Results:

- There are no findings of processes that run with system privileges.
- System settings have been found correctly set to ensure that no processes will run with system privileges.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions
- Test result (Passed or not)

4.3.4.3 No unused HTTP methods

Requirement Name: Unused HTTP methods shall be deactivated.

Requirement Description:

HTTP methods that are not required shall be deactivated. Standard requests to web servers only use GET, HEAD, and POST. If other methods are required, they shall not introduce security leaks such as TRACK or TRACE.

Test Case: TBA

***Test Name*:** TC_NO_UNUSED_HTTP_METHODS

Purpose:

Verify that the Web server has deactivated all HTTP methods that are not required.

Procedure and execution steps

Pre-Conditions:

- The tester has needed administrative privileges.
- A tester machine is available.
- Recommended: an automatic assessment tool has been configured / script adapted in line with the Requirement Description.

Execution Steps

- Check that relevant system settings and configurations are correct to ensure fulfilment of the requirement.

Expected Results:

- System settings and configurations have been found adequately set, in all Web components of the system, to ensure that unneeded HTTP methods are deactivated.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions
- Test result (Passed or not)

4.3.4.4 No unused add-ons

Requirement Name: Any add-ons and components that are not required shall be deactivated.

Requirement Description: All optional add-ons and components of the web server shall be deactivated if they are not required. In particular, CGI or other scripting components, Server Side Includes (SSI), and WebDAV shall be deactivated if they are not required.

Test Case:

***Test Name*:** TC_NO_UNUSED_ADD-ONS

Purpose:

To verify that the Web server has deactivated unneeded add-ons and unneeded scripting components.

Procedure and execution steps

Pre-Conditions:

- The vendor has supplied a list of add-ons or scripting tools for Web server components needed for system operation, and that therefore need to be exempted from the test investigation.

- The tester has administrative privileges.

- A tester machine is available.

- Recommended: an automatic assessment tool has been configured / script adapted in line with the Requirement Description.

Execution Steps

1. Check that the web server is only running and listening on known ports (e.g. tcp port 80 and/or 443). Check that CGI or other scripting components, Server Side Includes (SSI), and WebDAV are deactivated if they are not required. See also guidance under 4.3.4.12.

2. Check that nothing else has been installed than the web server.

3. Check that relevant system settings and configurations are correct to ensure fulfilment of the requirement.

Expected Results:

- System settings and configurations have been found adequately set, in all Web components of the system, to ensure that all unneeded add-ons or script components are deactivated.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions.

- Test result (Passed or not).

4.3.4.5 No compiler, interpreter, or shell via CGI or other server-side scripting

Requirement Name: No compiler, interpreter, or shell via CGI or other server-side scripting.

Requirement Description: If CGI (Common Gateway Interface) or other scripting technology is used, the CGI directory - or other corresponding scripting directory - shall not include compilers or interpreters (e.g. PERL interpreter, PHP interpreter/compiler, Tcl interpreter/compiler or operating system shells).

Test Case:

***Test Name*:** TC_NO_COMPILER_FOR_CGI

Purpose:

To verify that the Web server has deactivated unneeded add-ons and unneeded scripting components.

Procedure and execution steps

Pre-Conditions:

- The tester has administrative privileges

- A tester machine is available.

- Recommended: an automatic assessment tool has been configured /script adapted in line with the Requirement Description.

Execution Steps

1. Check that there are no compilers or interpreters (e.g., PERL interpreter, PHP interpreter/compiler, Tcl interpreter/compiler or operating system shells) in the directory/directories used for CGI or for other scripting tools (including PERL, PHP, and others).
2. Check that relevant system settings and configurations are correct to ensure fulfilment of the requirement.

Expected Results:

- System settings and configurations have been found adequately set, in all Web components of the system, to ensure that all unneeded add-ons or script components are deactivated.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions
- Test result (Passed or not)

4.3.4.6 No CGI or other scripting for uploads

Requirement Name: No CGI or other scripting for uploads.

Requirement Description: If CGI or other scripting technology is used, the associated CGI/script directory shall not be used for uploads.

Test Case:

Test Name: TC_NO_CGI_OR_SCRIPTING_FOR_UPLOADS

Purpose:

To test whether the upload directory is equal to the CGI/Scripting directory.

Procedure and execution steps:

Pre-Condition:

If the web server is configured with CGI/Scripting on, this test applies.

Execution Steps

Execute the following steps:

The tester checks whether the upload directory is configured to be different from the CGI/Scripting directory.

Expected Results:

The configured upload directory is different from the CGI/Scripting directory.

Additional evidence might be provided that shows that the web server has no write rights for the CGI/Scripting directory.

Expected format of evidence:

A part of the configuration file / screenshot of the configuration showing that the web server is properly configured.

4.3.4.7 No execution of system commands with SSI

Requirement Name: No execution of system commands with SSI.

Requirement Description: If Server Side Includes (SSI) is active, the execution of system commands shall be deactivated.

Test Case:

Test Name: TC_NO_EXECUTION_OF_SYSTEM_COMMANDS

Purpose:

To test whether it is possible to use the exec directive and if so, whether it can be used for system commands.

Procedure and execution steps:

Pre-Condition:

If the web server is configured with SSI active, this test applies.

Execution Steps

Execute the following steps:

The tester checks whether execution of system commands is disabled in the web server configuration.

Expected Results:

For example, a configuration file that shows that the IncludesNOEXEC (APACHE) or ssiExecDisable (IIS) is set.

Expected format of evidence:

A part of the configuration file / screenshot of the configuration showing that the web server is properly configured.

4.3.4.8 Access rights for web server configuration

Requirement Name: Access rights for web server configuration files shall only be granted to the owner of the web server process or to a user with system privileges.

Requirement Description: Access rights for web server configuration files shall only be granted to the owner of the web server process or to a user with system privileges. Implementation example: Delete "read" and "write" access rights for "others." Only grant "write" access to the user who configures the web server.

Test Case:

***Test Name*:** TC_ACCESS_RIGHTS_WEB_SERVER_FILES

Purpose:

To verify that the access rights for Web server configuration files are correctly set.

Procedure and execution steps

Pre-Conditions:

- The tester has administrative privileges
- A tester machine is available.
- Recommended: an automatic assessment tool has been configured / script adapted in line with the Requirement Description.

Execution Steps

- Check the access rights settings for Web server system configuration files.
- Check that relevant system settings and configurations are correct to ensure fulfilment of the requirement.

Expected Results:

- Access rights for system configuration files are adequately set.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions
- Test result (Passed or not)

4.3.4.9 No default content

Requirement Name: Default content shall be removed.

Requirement Description: Default content (examples, help files, documentation, aliases) that is provided with the standard installation of the web server shall be removed.

Test Case:

***Test Name*:** TC_NO_DEFAULT_CONTENT

Purpose:

To verify that there is no default content on the web server, that is not needed for web server operation, since such default content can be useful for an attacker.

Procedure and execution steps

Pre-Conditions:

- The tester has needed administrative privileges
- A tester machine is available.
- Recommended: an automatic assessment tool has been configured / script adapted in line with the Requirement Description.

Execution Steps

1. Check that all default content (examples, help files, documentation, aliases) that is provided with the standard installation of the web server has been removed.

Expected Results:

- No default content (examples, help files, documentation, aliases, un-needed directories or manuals) has been found to remain on any Web server component.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions.
- Test result (Passed or not).

4.3.4.10 No directory listings

Requirement Name: No directory listings / Directory Browsing.

Requirement Description: Directory listings (indexing) / "Directory browsing" shall be deactivated.

Test Case:

***Test Name*:** TC_NO_DIRECTORY_LISTINGS

Purpose:

To verify that Directory listings / Directory browsing has been deactivated in all Web server components.

Procedure and execution steps

Pre-Conditions:

- The tester has administrative privileges
- A tester machine is available.
- Recommended: an automatic assessment tool has been configured / script adapted in line with the Requirement Description.

Execution Steps

- Check that Directory listings (indexing) / "Directory browsing" has been deactivated in all Web server components.

Expected Results:

- Evidence that Directory listing / Directory browsing has been deactivated in all Web server components.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions
- Test result (Passed or not)

4.3.4.11 Web server information in HTTP headers

Requirement Name: Information about the web server in HTTP headers shall be minimized.

Requirement Description: The HTTP header shall not include information on the version of the web server and the modules/add-ons used.

Test Case:

***Test Name*:** TC_NO_WEB_SERVER_HEADER_INFORMATION

Purpose:

To verify that HTTP headers do not include information on the version of the web server and the modules/add-ons used.

Procedure and execution steps

Pre-Conditions:

- The tester has administrative privileges
- A tester machine is available.
- Recommended: an automatic assessment tool has been configured / script adapted in line with the Requirement Description.

Execution Steps

1. Check that HTTP headers do not include information on the version of the web server and the modules/add-ons used.

Expected Results:

- Evidence that HTTP headers do not include information on the version of the web server and the modules/add-ons used.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions
- Test result (Passed or not)

4.3.4.12 Web server information in error pages

Requirement Name: Web server information in error pages shall be deleted.

Requirement Description: User-defined error pages shall not include version information about the web server and the modules/add-ons used. Error messages shall not include internal information such as internal server names, error codes, etc. Default error pages of the web server shall be replaced by error pages defined by the vendor.

Test Case:

***Test Name*:** TC_NO_WEB_SERVER_ERROR_PAGES_INFORMATION

Purpose:

To verify that error pages and error messages do not include information about the web server.

Procedure and execution steps

Pre-Conditions:

- The tester has needed administrative privileges.
- A tester machine is available.
- Recommended: an automatic assessment tool has been configured / script adapted in line with the Requirement Description.

Execution Steps

- Check that generated error pages and error messages do not include information about the web server.

Expected Results:

- Evidence that generated error pages and error messages do not include information about the web server.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions
- Test result (Passed or not)

4.3.4.13 Minimized file type mappings

Requirement Name: File type- or script-mappings that are not required shall be deleted.

Requirement Description: File type- or script-mappings that are not required shall be deleted, e.g. php, phtml, js, sh, csh, bin, exe, pl, vbe, vbs.

Test Case:

***Test Name*:** TC_NO_WEB_SERVER_FILE_TYPE MAPPINGS

Purpose:

To verify that file type- or script-mappings that are not required have been deleted.

Procedure and execution steps

Pre-Conditions:

- The tester has needed administrative privileges.
- A tester machine is available.
- Recommended: an automatic assessment tool has been configured / script adapted in line with the Requirement Description.

Execution Steps

- Check that all file type- or script-mappings that are not required have been deleted.

Expected Results:

- Evidence that all file type- or script-mappings, that are not required, have been deleted.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- Log files and screen shots of test executions
- Test result (Passed or not)

4.3.4.14 Restricted file access

Requirement Name: The web server shall only deliver files which are meant to be delivered.

Requirement Description: Restrictive access rights shall be assigned to all files which are directly or indirectly (e.g. via links or in virtual directories) in the web server's document directory. In particular, the web server shall not be able to access files which are not meant to be delivered.

Test Case:

Test Name: TC_RESTRICTED_FILE_ACCESS

Purpose:

To test whether the restrictive access rights are assigned to all files which are directly or indirectly in the web server's document directory and to verify whether path traversal is made improbable.

Procedure and execution steps:

Pre-Condition:

1. The web server is configured according to the manual

Execution Steps

Execute the following steps:

1. The tester verifies that access rights on the servable content (meaning directories and files) is set to the following:
    a. The files are owned by the user that runs the web server;
    b. The files are not writable to others, except the web server's account;
2. The tester verifies that the user running the web server is an unprivileged account;
3. For Operating Systems that have chrooted environments, the tester verifies that the web server runs inside a jail or chrooted environment.

Expected Results:

- Name of user running the web server with the privileges of the account;
- Access rights of files and directories that the web server serves;
- Configuration that shows that the web server is in a chrooted environment.

Expected format of evidence:

A part of the configuration file / screenshot of the configuration showing that the web server, the file access rights and the account running the web server is properly configured.

4.3.4.15 Void

4.3.5 Network Devices

4.3.5.1 Traffic Separation

Requirement Name: Traffic Separation

Requirement Description:

The network product shall support physical or logical separation of traffic belonging to different network domains. For example, O&M traffic and control plane traffic belong to different network domains. See RFC 3871 [3] for further information.

Security Objective references: tba.

Test case:

Test Name: TC_TRAFFIC_SEPARATION

Purpose:

To test whether traffic belonging to different network domains is separated.

Procedure and execution steps:

Pre-Condition:

NOTE: This test applies if the network product is meant to handle traffic from different network domains, e.g. both O&M and control plane traffic.

The network product has at least two separate (logical) interfaces dedicated to different network domains. Network products for which the test applies and that fail to meet this precondition fail the test by definition.

Execution Steps

Execute the following steps:

1. The tester checks whether the network product refuses traffic intended for one network domain on all interfaces meant for the other network domain, and vice versa.

2. Step 1 is to be performed for all pairs of different network domains.

Expected Results:

The two tests should be successful.

Expected format of evidence:

A PASS or FAIL.

4.3.6 Network Functions in service-based architecture

4.3.6.1 Introduction

The purpose of the sub-clauses in 4.3.6 is to identify and describe the hardening related requirements for all Network Function (NF) within the 5G Core (5GC) utilizing Service-Based Interfaces (SBI) and the corresponding test cases.

4.3.6.2 No code execution or inclusion of external resources by JSON parsers

Requirement Name: No code execution or inclusion of external resources by JSON parsers.

Requirement Description:

Parsers used by Network Functions (NF) shall not execute JavaScript or any other code contained in JSON objects received on Service Based Interfaces (SBI). Further, these parsers shall not include any resources external to the received JSON object itself, such as files from the NF's filesystem or other resources loaded externally.

Threat References: TR 33.926 [4], clause 6.3.2.1, JSON Parser Exploits

Test Case:

Test Name: TC_JSON_PARSER_CODE_EXEC_INCL

Purpose:

NFs implementing SBI transfer application data serialized as JSON objects. When receiving such data, an NF parses this JSON representation and creates equivalent internal data structures. Since the contents of the JSON objects must be considered untrusted, blindly executing code fragments or loading resources from a local path or Uniform Resource Identifier (URI) must not be possible.

Procedure and execution steps:

Pre-Conditions:

- The tester has the privileges to log in the network product and to access to the all system resources (e.g. log files)
- A list of all available network services containing at least the following information shall be included in the documentation accompanying the Network Product:
  - all interfaces providing IP-based protocols;
  - the available transport layer protocols on these interfaces;
  - their open ports and associated services in the form of an OpenAPI3.0 interface specification;
- The tester should have access to an effective Web Application Security (WAS) test tool that allows to generate HTTP messages exploiting JSON parsers that do not prevent the above-mentioned scenarios of code execution

and loading external resources. The accredited test lab is expected to have sufficient expertise to recognize the level of effectiveness of the available tools.

- A network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product and on a tester machine is available.

Execution Steps

1. Execution of available WAS test tools against the network product's API endpoints via its Service Based Interfaces.
2. Using a network traffic analyser on the network product, e.g. TCPDUMP or an external traffic analyser directly connected to the network product, the tester verifies that no external resources get loaded during JSON parsing.
3. Depending on the actual JavaScript code in the HTTP message, the tester verifies that the network product does not execute any of the contained actions.

Expected Results:

- The NF does not load any resources external to the JSON object itself.
- The NF does not execute any JavaScript code contained in JSON objects.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The used tool(s) name and version information
- Settings and configurations used
- The output log file of the chosen tool that displays the results (passed/failed).
- Screenshot
- Test result (Passed or not)

4.3.6.3 Unique key values in IEs

Requirement Name: Validation of the unique key values in IEs.

Requirement Reference: 3GPP TS 29.501 Principles and Guidelines for Services Definition [13], clause 6.2.

Requirement Description: "For data structures where values are accessible using names (sometimes referred to as keys), e.g. a JSON object, the name shall be unique. The occurrence of the same name (or key) twice within such a structure shall be an error and the message shall be rejected".

Threat References: TR 33.926 [4], clause 6.3.2.2, JSON Parser not Robust

Test Case:

NOTE: This requirement can also be verified as part of Robustness and Protocol fuzzing tests as defined in clause 4.4.4 Robustness and fuzz testing according to referenced requirements.

Purpose:

Verify that the API implementation fullfills the requirements as specified in 29.501 [13], clause 6.2.

Pre-Conditions:

Test environment with network product under test. Rest of the network and network products may be simulated.

Execution Steps

1) The test equipment sends requests with duplicate keys in message IE payload to the network product under test.

2) The test equipment sends valid requests to network product under test

Expected Results:

1) Network product under tests responses with an error message

2) Network product under test still responses normally to valid requests

Expected format of evidence:

- A testing report provided by the testing agency which will consist of the following information:
  - The used tool(s) name and version information,
  - Settings and configurations used
  - The output log file of the chosen tool that displays the results (passed/failed).
  - Test result (Passed or not)
  - Log/evidence tracing possible crashes
  - Information of any input causing unspecified, undocumented, or unexpected behaviour

4.3.6.4 The valid format and range of values for IEs

Requirement Name: Validation of the IEs limits.

Requirement Reference: 3GPP TS 29.501 Principles and Guidelines for Services Definition [13], clause 6.2

Requirement Description: "The valid format and range of values for each IE, when applicable, shall be defined unambiguously:

- For each message the number of leaf IEs shall not exceed 16000.
- The maximum size of the JSON body of any HTTP request shall not exceed 2 million bytes.
- The maximum nesting depth of leaves shall not exceed 32."

Threat References: TR 33.926 [4], clause 6.3.2.2, JSON Parser not Robust

Test Case:

NOTE: This requirement can also be verified as part of Robustness and Protocol fuzzing tests as defined in clause 4.4.4 Robustness and fuzz testing according to referenced requirements.

Purpose:

Verify that the API implementation fullfills the requirements as specified in 29.501[13], clause 6.2.

Pre-Conditions:

Test environment with network product under test. Rest of the network may be simulated.

Execution Steps

1) The test equipment sends requests with out of bounds IEs towards the network product under test.

Expected Results:

- Network product under tests responses with an error message.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The used tool(s) name and version information,
- Settings and configurations used.
- The output log file of the chosen tool that displays the results (passed/failed).
- Test result (Passed or not).
- Log/evidence tracing possible crashes.
- Information of any input causing unspecified, undocumented, or unexpected behaviour.

4.4 Basic vulnerability testing requirements

4.4.1 Introduction

Basic Vulnerability Testing activities consist of requirements for running automated Free and Open Source Software (FOSS) and Commercial off-the-shelf (COTS) security testing tools against the external interfaces of a Network Product. These activities cover at least four aspects: Port Scanning, Vulnerability Scanner by the use of Vulnerability scanners and robustness/fuzz testing, and endpoint scanning. For each of these aspects, test requirements and test results are described in the present clause.

NOTE: The individual tools used for Basic Vulnerability Testing are selected by the evaluator. The SECAM accreditation body will ensure during accreditation of the evaluator's laboratory that the testers are able to utilize adequate tools.

4.4.2 Port Scanning

Requirement Name: Port scaning

Requirement Description:

It shall be ensured that on all network interfaces, only documented ports on the transport layer respond to requests from outside the system.

The test for this requirement can be carried out using a suitable tool or manually performed as described below. If a tool is used then the tester needs to provide evidence, e.g. by referring to the documentation of the tool, that the tool actually provides functionality equivalent to the steps described below.

Test Case:

Test Name: TC_BVT_PORT_SCANNING

Purpose:

To ensured that on all network interfaces, only documented ports on the transport layer respond to requests from outside the system

Procedure and execution steps:

Pre-Conditions:

A list of all available network services containing at least the following information shall be included in the documentation accompanying the Network Product:

1. all interfaces providing IP-based protocols;

2. the available transport layer protocols on these interfaces;
3. their open ports and associated services per transport layer protocol;
4. and a free-form description of their purposes.

The port scanning tool that is used shall be capable to detect open ports on the relevant transport layer protocols.

NOTE: It might not be possible for certain transport layer protocols (like UDP) to unambiguously detect whether a port is open or not by means of external port scanning. Also in some circumstances it might not be efficient to do external port scanning, e.g. if there are security measures to limit the rate a system can be probed. In those cases the accredited evaluator's test laboratory determines another means suitable to verify which ports are open.

Execution Steps

The accredited evaluator's test lab is required to execute the following steps:

1. Verification of the compliance to the prerequisites:
   a. Verification that the list of available network services is available in the documentation of the Network Product
   b. Validation that all entries in the list of services are meaningful and reasonably necessary for the operation of the Network Product class
2. Identification of the open ports by means of capable port scanning tools or other suitable testing means
3. Verification that the list of identified open ports matches the list of available network services in the documentation of the Network Product

Expected Results:

The used tool(s) name, their unambiguous version (also for plug-ins if applicable), used settings, and the relevant output containing all the technically relevant information about test results is evidence and shall be part of the testing documentation.

All discrepancies between the list of identified open ports and the list of available network services in the documentation shall be highlighted in the testing documentation.

Expected format of evidence:

Output of portscan and list of identified discrepancies.

4.4.3 Vulnerability scanning

Requirement Name: Vulnerability scanning

Requirement Description:

The purpose of vulnerability scanning is to ensure that there are no known vulnerabilities (or that relevant vulnerabilities are identified and remediation plans in place to mitigate them) on the Network Product, both in the OS and in the applications installed, that can be detected by means of automatic testing tools via the Internet Protocol enabled network interfaces.

Vulnerability scanning tools may also report false positives and they shall be investigated and documented in the test report.

The test for this requirement can be carried out using a suitable tool or manually performed as described below. If a tool is used then the tester needs to provide evidence, e.g. by referring to the documentation of the tool, that the tool actually provides functionality equivalent to the steps described below.

Test case:

Test Name: TC_BVT_VULNERABILITY_SCANNING

Purpose:

The purpose of vulnerability scanning is to ensure that there are no known vulnerabilities (or that relevant vulnerabilities are identified and remediation plans in place to mitigate them) on the Network Product that can be detected by means of automatic testing tools via the Internet Protocol enabled network interfaces.

Procedure and execution steps:

Pre-Conditions:

A list of all available network services containing at least the following information shall be included in the documentation accompanying the Network Product:

- all interfaces providing IP-based protocols;
- the available transport layer protocols on these interfaces;
- their open ports and associated services;
- and a free-form description of their purposes.

NOTE 1: This list is to be validated as part of the BVT port scanning activity.

The used vulnerability scanning tool shall be capable to detect known vulnerabilities on common services. The used vulnerability information shall be reasonably recent at the time of testing.

Execution Steps

The accredited evaluator's test lab is required to execute the following steps:

1. Execution of the suitable vulnerability scanning tool against all interfaces providing IP-based protocols of the Network Product.
2. Evaluation of the results based on their severity.

Expected Results:

The used tool(s) name, their unambiguous version (also for plug-ins if applicable), used settings, and the relevant output is evidence and shall be part of the testing documentation.

The discovered vulnerabilities (including source, example CVE ID), together with a rating of their severity, shall be highlighted in the testing documentation.

COTS Vulnerability scanners, by their nature, (e.g. depending on how they are configured) may result in false findings/positives. The tool’s documentation may even mention that the failing test shall be repeated to check whether it is really a recurring problem or not. The tester shall make best effort to determine if there is an issue with NE or the test tool and if necessary, work with the vendor of the network product to come to a consensus on the test result outcome.

NOTE 2: This testing documentation is input to the vulnerability mitigation process (that may include patching). This is part of the product lifecycle management process developed by GSMA SECAG.

Expected format of evidence:

Output of BVT tool.

4.4.4 Robustness and fuzz testing

Requirement Name: Robustness and fuzz testing

Requirement Reference: 4.2.6.2.2. – Interface Robustness requirements

Requirement Description:

It shall be ensured that externally reachable services are reasonably robust when receiving unexpected input

Test case:

Test Name: TC_BVT_ROBUSTNESS AND FUZZ TESTING

Purpose:

To verify that the network product provides externally reachable services which are robust against unexpected input. The target of this test are the protocol stacks (e.g. diameter stack) rather than the applications (e.g. web app).

Procedure and execution steps:

Pre-Conditions:

- The tester has the privileges to log in the network product and to access all system resources (e.g. log files)

- A list of all available network services containing at least the following information shall be included in the documentation accompanying the Network Product:

- all interfaces providing IP-based protocols;

- the available transport layer protocols on these interfaces;

- their open ports and associated services;

- and a free-form description of their purposes.

NOTE: This list is to be validated as part of the BVT port scanning activity.

- The robustness and fuzzing tools that are selected for this test shall utilize state-of-the-art technology to identify input which causes the Network Product to behave in an unspecified, undocumented, or unexpected manner.

- Fuzz testing tools are a highly sophisticated technology and adaptation to the individual protocols in question is needed to be effective. Therefore, there is a lack of available effective fuzz testing tools available especially for protocols proprietary to the Telco industry. Taking into account note 4 of TR 33.916's clause 7.2.4, test labs shall acquire fuzz testing tools for those protocols where commercially feasible.

- It needs to be taken into account that fuzz testing tools might show drastic differences in terms of effectiveness. The accredited test lab is expected to have sufficient expertise to recognize the level of effectiveness of the available tools.

- A network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product and on a tester machine is available.

Execution Steps

The accredited evaluator's test lab is required to execute the following steps:

1. Execution of available effective fuzzing tools against the protocols available via interfaces providing IP-based protocols of the Network Product for an amount of time sufficient to be effective.

2. Execution of available effective robustness test tools against the protocols available via interfaces providing IP-based protocols of the Network Product for an amount of time sufficient to be effective.

3. For both step 1 and 2:

    a. Using a network traffic analyser on the network product (e.g. TCPDUMP) or an external traffic analyser directly connected to the network product, the tester verifies that the packets are correctly processed by the network product.

    b. The testers verifies that the network product and any running network service does not crash.

    c. The execution of tests shall run sufficient times.

Expected Results:

A list of all of the protocols of the network product reachable externally on an IP-based interface, together with an indication whether effective available robustness and fuzz testing tools have been used against them, shall be part of the testing documentation. If no tool can be acquired for a protocol, a free form statement should explain why not.

The used tool(s) name, their unambiguous version (also for plug-ins if applicable), used settings, and the relevant output is evidence and shall be part of the testing documentation.

Any input causing unspecified, undocumented, or unexpected behaviour, and a description of this behaviour shall be highlighted in the testing documentation.

COTS fuzzing tools, by their nature, may have an acceptable failure rate (e.g. 0.1%) due to different non-deterministic variables in their implementation. At some point the tool's documentation may even mention that the failing test shall be repeated to check whether it is really a recurring problem or not. The tester shall make best effort to determine if there is an issue with NE or the test tool and if necessary, work with the vendor of the network product to come to a consensus on the test result outcome.

Expected format of evidence:

A testing report provided by the testing agency which will consist of the following information:

- The used tool(s) name and version information,
- Settings and configurations used
- The output log file of the chosen tool that displays the results (passed/failed).
- Screenshot
- Test result (Passed or not)
- Log/evidence tracing possible crashes
- Any input causing unspecified, undocumented, or unexpected behaviour

Annex A (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
12-2016	SA#74	SP-160781	0001	1	F	Detailing clauses 4.2.1, 4.2.3.1 and 4.3.1	14.1.0
06-2017	SA#76	SP-170426	0003	-	F	Resolution of editor's notes in 33.117	14.2.0
2018-03	SA#79	SP-180044	0004	1	F	Collection of changes based on feedback from GSMA SECAG	14.3.0
2018-06	-	-	-	-	-	Update to Rel-15 version (MCC)	15.0.0
2018-09	SA#81	SP-180698	0006	1	A	Clarifications for section 4.2.3.2.2 and 4.3.2.1	15.1.0
2018-12	SA#82	SP-181028	0012	-	F	Update the requirement in 4.2.3.2.2	15.2.0
2018-12	SA#82	SP-181028	0014	-	F	Formatting issue	15.2.0
2018-12	SA#82	SP-181028	0015	-	F	Adding missing references in TS 33.117	15.2.0
2018-12	SA#82	SP-181027	0008	-	B	5G inclusion in TS 33.117	16.0.0
2018-12	SA#82	SP-181027	0011	-	D	Editorial corrections in TS 33.117	16.0.0
2018-03	SA#83	SP-190102	0020	-	A	Protection from buffer overflows	16.1.0
2018-03	SA#83	SP-190102	0030	-	A	New proposal on the length of password and other clarifications.	16.1.0
2019-09	SA#85	SP-190677	0034	1	A	Corrections on IP packet forwarding	16.2.0
2019-09	SA#85	SP-190677	0035	1	A	Clarification on fuzz test run	16.2.0
2019-09	SA#85	SP-190677	0038	1	A	Clarification on the intention of the requirement	16.2.0
2019-09	SA#85	SP-190677	0041	1	A	A document is needed to show the support features in 33.210 and 33.310	16.2.0
2019-09	SA#85	SP-190677	0044	1	A	Align account numbers in testcase with the requirement	16.2.0
2019-09	SA#85	SP-190688	0047	1	B	Addition of General SBA/SBI aspects in TS 33.117	16.2.0
2019-09	SA#85	SP-190677	0052	-	A	Clarification on test cases in TR 33.117	16.2.0
2019-09	SA#85	SP-190677	0054	-	A	Update testcase of 4.2.4.1.1.2 and 4.2.4.1.1.3	16.2.0
2019-12	SA#86	SP-191138	0055	-	D	Miscellaneous Editorial clarifications	16.3.0
2019-12	SA#86	SP-191138	0056	-	F	Adding abbreviations and corrections for alignment	16.3.0
2020-03	SA#87E	SP-200136	0057	1	F	Clarification on the clause 4.3.6.4 according to TS 29.501	16.4.0
2020-03	SA#87E	SP-200136	0058	1	F	Clarification on PLMN ID verification	16.4.0
2020-03	SA#87E					Correction of version in history table	16.4.1
2020-07	SA#88E	SP-200357	0059	-	A	Update the clause 4.2.6.2.4 (R16)	16.5.0
2020-07	SA#88E	SP-200358	0062	1	F	One more clarification about system handling during overload situations	16.5.0
2020-07	SA#88E	SP-200358	0063	1	F	Modification for policy regarding consecutive failed login attempts	16.5.0
2020-07	SA#88E	SP-200358	0064	-	F	Password structure clarification	16.5.0
2020-07	SA#88E	SP-200358	0065	-	F	Clarification on the examples of the delay	16.5.0
2020-12	SA#90e	SP-201004	0066	1	F	Correction of test case for access token verification failure handling in different PLMN	16.6.0
2021-06	SA#92e	SP-210446	0069	-	F	Clarification on external file system mount restrictions	16.7.0
2021-06	SA#92e	SP-210446	0072	-	B	CR to include new test cases and fix editorial in 33.117	17.0.0
2022-09	SA#97e	SP-220887	0076	-	A	Clarification on Execute rights exclusive for CGI/Scripting directory	17.1.0
2022-09	SA#97e	SP-220887	0078	1	A	Clarification on Handling of ICMP	17.1.0

History

Document history		
V17.0.0	May 2022	Publication
V17.1.0	September 2022	Publication