ETSI GR NFV-SEC 009 V1.3.1 (2025-01)

Network Functions Virtualisation (NFV);
NFV Security;
Report on use cases and technical approaches for multi-layer host administration

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
RGR/NFV-SEC009ed131

Keywords
administration, regulation, security

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

Contents

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

Foreword

This Group Report (GR) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

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

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

Introduction

The Security Problem Statement, ETSI GS NFV-SEC 001 [i.1] identifies an issue with multi-layer administration for NFV. Multi-layer administration seeks to provide methods, capabilities, procedures and assurances that safeguard Virtual Machines or Containers running on a virtualisation host from interference. The specific problem is that any user or process with root access to the hosting service can normally view and change the memory and processes of any hosted application. This is due to the fact that in the default administrative configuration for the majority of host-based virtualisation systems - whether using hypervisors or Containers - any process or administrator operating at the "base" level has access to the memory of all applications - including VMs and Containers - running on that host. The term *inspection* is often used to refer to the ability for processes to directly interact with system memory. Further detail is provided in clause 6.1.1.

Although this configuration is generally acceptable when the hosted applications and the hosting service operate in the same trust domain, or when the hosted applications are in the same trust context and a subordinate trust domain to the hosting service, there are a number of use cases where the trust relationship from the hosted application to the hosting service does not conform to this model. In these cases, the hosted application may wish to protect a set of its resources from the hosting service.

Note that there are also attacks in the opposite direction: from the hosted application against the hosting service. While serious, these are well understood issues and most hosting services already track vulnerabilities in this context and provide defensive measures against these types of attacks. Another type of attack is from one hosted application against another hosted application on the same hosting service. Neither of these "top-down" attacks are considered explicitly in the present document, however, some of the methods and techniques presented here will reduce the incidence of such attacks (e.g. hardware mediated secure enclaves). The focus of the present document, then, is on securing hosted applications against attacks by the hosting service, as well as limiting undesired visibility.

Note that multi-layer administration in the context of NFV should not be confused with the similar term "Multi-Layer Security" (MLS), though certain concepts relevant to MLS may be relevant or referenced in the present document.

1 Scope

The present document addresses multi-layer administration use cases and technical approaches, an issue identified in the Security Problem Statement, ETSI GS NFV-SEC 001 [i.1]. Multi-layer administration seeks to provide methods, capabilities, procedures and assurances - of various strengths based on requirements and available technologies and techniques - that safeguard Virtual Machines or Containers running on a virtualisation host ("hosted applications") - from interference (of various types) by the host system or platform ("hosting service").

The scope of the present document is generally the system comprising the hosting service, associated hardware (including TPM, GPU, etc.), software and configuration, and the hosted application. Some requirements and measures outside this context are also considered, but not necessarily in equal depth.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

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

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

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GS NFV-SEC 001: "Network Functions Virtualisation (NFV); NFV Security; Problem Statement".

[i.2] ETSI GS NFV-SEC 003: "Network Functions Virtualisation (NFV); NFV Security; Security and Trust Guidance".

[i.3] ETSI TR 103 331: "Cyber Security (CYBER); Structured threat information sharing".

[i.4] ETSI TS 102 232: "Lawful Interception (LI); Handover Interface for IP delivery".

[i.5] ETSI TS 101 331: "Lawful Interception (LI); Requirements of Law Enforcement Agencies".

[i.6] ETSI TS 102 656: "Lawful Interception (LI); Retained Data; Requirements of Law Enforcement Agencies for handling Retained Data".

[i.7] ETSI TS 102 657: "Lawful Interception (LI); Retained data handling; Handover interface for the request and delivery of retained data".

[i.8] ETSI GR NFV-SEC 007: "Network Functions Virtualisation (NFV); Trust; Report on Attestation Technologies and Practices for Secure Deployments".

[i.9] NIST Special Publication 800-122: "Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)".

[i.10] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. .

[i.11] TCG PC: "Client Specific Implementation Specification for Conventional BIOS - Specification Version 1.21 Errata".

[i.12] ETSI GS NFV-SEC 004: "Network Functions Virtualisation (NFV); NFV Security; Privacy and Regulation; Report on Lawful Interception Implications".

[i.13] Forensics Whitepapers.

[i.14] TCG TPM 2.0 Library: "Trusted Platform Module Library Specification, Family 2.0".

[i.15] TCG TSS 2.0 TAB and Resource Manager: "TSS TAB and Resource Manager Specification".

[i.16] NIST FIPS 140-2: "Security Requirements for Cryptographic Modules".

[i.17] TCG: "Virtualized Trusted Platform Architecture Specification".

[i.18] ETSI GS NFV-SEC 010: "Network Functions Virtualisation (NFV); NFV Security; Report on Retained Data problem statement and requirements".

[i.19] ETSI GS NFV 001: "Network Functions Virtualisation (NFV); Use Cases".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

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

AAA	Authentication, Authorisation & Auditing
ADMF	Administrative Function (for Lawful Interception)
API	Application Programming Interface
AUC	AUthentication Centre
BIOS	Basic Input/Output System
BMSC	Broadcast-Multicast Service Centre
BRAS	Broadband Remote Access Server
CA	Certificate Authority
CIA	Confidentiality, Integrity and Availability
CoT	Chain of Trust
CPU	Central Processing Unit
CRTM	Core Root of Trust for Measurement
CS	Circuit Switched
CSCF	Call Session Control Function
DMA	Direct Memory Access
DSLAM	Digital Subscriber Line Access Multiplexer
EMS	Element Management System
ESXi	Elastic Sky X integrated
FIPS	Federal Information Processing Standards
GGSN	Gateway GPRS support node
GMSC	Gateway Mobile Switching Centre
GPU	Graphics Processing Unit
HBRT	Hardware-Based Root of Trust
HLR	Home Location Register
HSM	Hardware Security Module
HSS	Home Subscriber Server
HW	Hardware

I/O	Input/Output
IaaS	Infrastructure as a Service
IBCF	Interconnection Border Control Function
ID	IDentifier
IMS	IP Multimedia Subsystem
IMS-ALG	IMS Application Level Gateway
KVM	KVM hypervisor software
LBA	Logical Block Array(s)
LEA	Law Enforcement Agency
LI	Lawful Interception
LTE	Long Term Evolution
MAC	Modify, Access, Create
MFRP	Multimedia Resource Function Processor
MME	Mobility Management Entity
MRFC	Media Resource Function Controller
MSC	Mobile Switching Centre
NFV	Network Function Virtualisation
NFVI	Network Function Virtualisation Infrastructure
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
OS	Operating System
OSS	Operations Support Systems
PC	Personal Computer
PCI	Payment Card Industry
PCR	Platform Configuration Register
P-CSCF	Proxy - Call Session Control Function
PDN-Gateway	Packet Data Network Gateway
PII	Personal Identifiable Information
PKI	Public Key Infrastructure
POI	Point Of Interception
pTPM	physical Trusted Platform Module
RAID	Redundant Array of Inexpensive Disks OR Redundant Array of Independent Disks
RAM	Random Access Memory
RD	Retained Data
RoT	Root of Trust
RSA	RSA encryption algorithm
RTM	Root-of Trust for Measurement
SBC	Session Border Controller
S-CSCF	Serving – Call Session Control Function
SDN	Software-Defined Networking
SED	Self-Encrypting Drive
SGSN	Serving GPRS support node
S-GW	Serving - GateWay
SLA	Service Level Agreement
SMSC	Short Message Service Centre
SW	Software
TAB	TPM Access Broker
TBB	Trusted Building Block
TCG	Trusted Computing Group

NOTE: See www.trustedcomputinggroup.org.

ToR	Top of Rack
TPM	Trusted Platform Module
UICC	Universal Integrated Circuit Card
vCPE	virtual Customer Premises Equipment
vHSM	virtual HSM
VIM	Virtual Infrastructure Manager
VLR	Visitor Location Register
VM	Virtual Machine
VMM	Virtual Machine Manager
VNF	Virtual Network Function

VNFCI	Virtual Network Function Component Instance
VNFI	Virtual Network Function Instance
VNFM	Virtual Network Function Manager
VoLTE	Voice over LTE
vTPM	virtual TPM.

4 Use cases for multi-layer administration

4.0 Use cases - introduction

These use case descriptions provide various levels of detail, some referencing items in clause 5.0.5 to note which specific requirements are relevant to the use case. This is particularly the case where regulatory requirements allow for detailed definition of requirements (in the case, for instance, of Lawful Interception clause 4.5 and Retained Data clause 4.6. In other cases, the specific requirements will depend more on the business requirements of the operator, the current business environment and local norms: here, a general description of the use case is provided, and some suggestions made.

4.1 Multi-tenant hosting

This is the case where one operator hosts VNFs (or VNFCIs) from one or more operators on NFVI owned and/or operated by the first operator. This is identified in ETSI GS NFV 001 [i.19] as Use Case #1 "Infrastructure as a Service", but is differentiated in the present document from the use case in clause 4.2.

The relevance to the present document is that the first operator (the hosting operator) may need to be able to provide assurances that sensitive processes, algorithms and data (e.g. subscriber details) owned by the other operators cannot be viewed or changed by the NFVI operator, whether intentionally or unintentionally. The exact requirements for such an agreement will be subject to contractual arrangements between different operators, and are not therefore considered in detail in the present document. They may include, however, the requirements described in the following clauses of the present document:

- 5.1.1 Capability assertion and attestation at boot-time.
- 5.1.3 Assert secure provision of hosted application.
- 5.1.6 Location assertion.
- 5.2.1 Confidentiality of data.
- 5.2.3 Confidentiality of processes.
- 5.2.6 Secure communications.
- 5.2.7 Secure storage.
- 5.2.8 Secure clean-up.
- 5.2.10 Assurance of compliance by hosting service.
- 5.2.11 Availability of entropy source.
- 5.3.2 Workload placement policy and operation security.
- 5.3.3 Availability of an attestation authority.

Another requirement that may arise is that of resource allocation: particularly the availability of sufficient CPU cycles and network bandwidth for hosted VNFs/VNFCIs. This is outside the scope of the present document, and it is expected that guarantees would be made by contractual agreements such as Service Level Agreements (SLAs). Monitoring for such SLAs is also considered outside the scope of the present document.

4.2 Infrastructure as a service (IaaS)

Infrastructure as a service ("IaaS") is the case where a service provider may wish to provide infrastructure services to third party with the extra guarantee that the service provider cannot view or change data or algorithms in the hosted applications. This is not considered a "pure" NFV use case, as it is more akin to data centre hosting than service provision, but is briefly considered here as it shares similar requirements with multi-tenant and the measures available are also applicable. Infrastructure as a service is also a service offered by other business units of operators, and it is expected that best practice should be shared in both directions. A key point here is that customers may have requirements to keep encryption keys safe (see example 1 or example 2), and it will therefore fall to the hosting service provider to ensure that measures are in place to service this requirement.

The exact requirements for such a service will depend on the services offered by the hosting service provider, and are not therefore considered in detail in the present document. They may include, however, the requirements described in the following clauses of the present document:

- 5.1.1 Capability assertion and attestation at boot-time
- 5.1.3 Assert secure provision of hosted application
- 5.1.6 Location assertion
- 5.2.1 Confidentiality of data
- 5.2.3 Confidentiality of processes
- 5.2.6 Secure communications
- 5.2.7 Secure storage
- 5.2.8 Secure clean-up
- 5.2.10 Assurance of compliance by hosting service
- 5.2.11 Availability of entropy source
- 5.3.2 Workload placement policy and operation security
- 5.3.3 Availability of an attestation authority

As with Multi-tenant hosting clause 4.1, another requirement that may arise is that of fair resource allocation: particularly the availability of sufficient CPU cycles and network bandwidth for hosted VNFs/VNFCIs. This is outside the scope of the present document, and it is expected that guarantees would be made by contractual agreements such as Service Level Agreements (SLAs). Monitoring for such SLAs is also considered outside the scope of the present document.

4.3 Security Sensitive Application Functions

4.3.1 Introduction

This use case concerns the segregation of sensitive application functions from other network functions, where restricted access and additional security domain separation requirements may be applied by operators.

Examples of such functions are the 3GPP AUC (master cryptographic key database responsible for holding UICC keys and generating authentication vectors) and the HSS which contains the 3GPP user subscription information. Typically operators allow a very restricted set of administrators access to such sensitive functions compared with other network elements.

These functions are considered to be largely standalone islands within an operator's network although they will interconnect with other VNFs in other administrative domains via specific restricted interfaces at the virtualised application layer.

4.3.2 Applicability of security requirements in the context of Sensitive Application Functions

This clause gives specific interpretation of clause 5 in the context of Sensitive Application Functions.

Table 1

Clause of the present document	Notes for Security Sensitive Application Functions
5.0.1 Overview	Depending on the Sensitive Function availability is likely to be important (e.g. HSS & AUC), as network and/or user services will not be available without these functions.
5.0.2 Prevention versus remediation	Prevention is likely more important than remediation as the network may not function without these functions. However if a negative security event cannot be prevented, remediation is very important.
5.1.1 Capability assertion and attestation at boot-time	Important for Sensitive Application Functions but no specific provisions are noted.
5.1.2 Capability assertion and attestation at run-time	Important for Sensitive Application Functions but no specific provisions are noted.
5.1.3 Assert secure provision of hosted application	Important for Sensitive Application Functions but no specific provisions are noted.
5.1.4 Assert own system integrity at boot	Important for Sensitive Application Functions. VNFI may be to verify integrity of databases, static configuration data and application root key chain (e.g. AUC).
5.1.5 Assert continued integrity of own system at run-time	Important for Sensitive Application Functions. Loss of integrity of Sensitive Functions may lead to loss of integrity of whole virtualised network and services.
5.1.6 Location assertion	Depends on specific Sensitive Function.
5.2.1 Confidentiality of data	Depends on specific Sensitive Function. Would be critical for functions containing application cryptographic functions (e.g. AUC) but may not be critical in all functions.
5.2.2 Confidentiality of data-related metadata	Depends on specific Sensitive Function. Would be critical for functions containing application cryptographic functions (e.g. AUC) or subscriber databases (e.g. HSS) but may not be critical in all functions.
5.2.3 Confidentiality of processes	Important for Sensitive Application Functions but no specific provisions are noted.
5.2.4 Confidentiality of process-related metadata	Important for Sensitive Application Functions but no specific provisions are noted.
5.2.5 Concealment of resource usage	May be desirable for some functions to prevent attacks on cryptographic functions but absolute concealment unlikely to be required.
5.2.6 Secure communications	Important for Sensitive Application Functions but no specific provisions are noted.
5.2.7 Secure storage	Important for Sensitive Application Functions. Storage of cryptographic keys, algorithms and other sensitive information will require secure storage.
5.2.8 Secure clean-up	Important for Sensitive Application Functions. Cryptographic keys, algorithms and customer data subject to Data Protection requirements will require secure clean-up.
5.2.9 Secure routing/switching	Important for Sensitive Application Functions but no specific provisions are noted.
5.2.10 Assurance of compliance by hosting service	Important for Sensitive Application Functions but no specific provisions are noted.
5.2.11 Availability of entropy source	May be important for some functions (e.g. AUC) but not for other.
5.3.1 Secure routing/switching	Important for Sensitive Application Functions but no specific provisions are noted.
5.3.2 Workload placement policy and operation security	Important for Sensitive Application Functions but no specific provisions are noted.
5.3.3 Availability of an attestation authority	May be important for some functions.

4.3.3 Notes on the technologies and measures in the context of Sensitive Application Functions

The following notes give specific interpretation of clause 6 in the context of Sensitive Application Functions.

Table 2

Clause of the present document	Notes for Sensitive Application Functions
6.1 Memory inspection	Memory inspection by a hosting service would cause issues for some Sensitive Functions (e.g. AUC). In general hosting services may not be trusted to introspect Sensitive Function details on hosted services.
6.2 Secure logging	There is a requirement for capability of logging for Sensitive Application Functions. However depending on the function, these logs may be need to be treat separately from other functions.
6.3 OS-level access control and 6.4 Post-incident analysis	This is important, but no specific provisions are noted.
6.5 Physical controls and alarms and 6.6 Personnel controls and checks	This is important and will depend on the specific function (e.g. AUC), but no specific provisions are noted.
6.8 Read-only partitions and 6.9 Write-only partitions	Read or write-only partitions will be required by some Sensitive Application Functions.
6.11 Communications Security	Confidentiality and integrity of network traffic are critical for most Sensitive Functions.
6.12 Measured boot, 6.13 Secured boot	This is important, but no specific provisions are noted.
6.14 Constant resource usage	Unlikely to be necessary in most sensitive functions. However may be required for specific components of cryptographic and similar functions (e.g. AUC).
6.15 Attestation, 6.16 Hardware-mediated execution enclaves, 6.17 Trusted Platform Module (TPM)	TPMs or equivalent implementations that include the protection capabilities, as provided by TPMs, may be mandated to be hardware based. Virtual modules may not be sufficiently robust.
6.18 Self-encrypting drives/storage	Confidentiality and integrity of data at rest is critical for most Sensitive Functions.
6.19 Direct Memory Access to hardware resources	Unlikely to be necessary in most sensitive functions but some VNFs are likely to require access to hardware accelerator functional (e.g. specialist cryptographic functions - see, for example, clause 6.20).
6.20 Hardware Security Modules	Some VNFs may require access to the security capabilities offered by HSMs.
6.21 Software integrity protection and verification	Software integrity protection and verification is expected to be required for almost all use cases.

4.4 Security Network Monitoring & Control Functions

4.4.1 Introduction

This use case concerns the segregation of monitoring functions from other network functions, where restricted access and additional security domain separation requirements may be applied by operators.

Examples of such functions are the routers, firewalls, packet monitoring filters or other network defensive functions (e.g. proxy servers). These functions are used logically to protect virtualised functions running in other VMs.

These functions may be grouped in a single administrative domain or multiple parallel domains each containing one or more functions (e.g. one or more firewalls).

4.4.2 Applicability of security requirements in the context of Network Monitoring & Control Functions

The following notes give specific interpretation of clause 5 in the context of Network Monitoring & Control Functions.

Table 3

Clause of the present document	Notes for Network Monitoring & Control Functions
5.0.1 Overview	Depending on the monitoring & control Function, availability is likely to be important (e.g. routers & firewalls), as network and/or user services will not be available without these functions.
5.0.2 Prevention versus remediation	Prevention is likely more important than remediation as the network may not function without these functions. However if a negative security event cannot be prevented, remediation is very important.
5.1.1 Capability assertion and attestation at boot-time	This is important, but no specific provisions are noted.
5.1.2 Capability assertion and attestation at run-time	This is important, but no specific provisions are noted.
5.1.3 Assert secure provision of hosted application	This is important, but no specific provisions are noted.
5.1.4 Assert own system integrity at boot	Given that these functions control access to other VNFs and are instrumental in defending the network, integrity of these functions is paramount.
5.1.5 Assert continued integrity of own system at run-time	Given that these functions control access to other VNFs and are instrumental in defending the network, integrity of these functions is paramount.
5.1.6 Location assertion	Given that these functions control access to other VNFs and are instrumental in defending the network, correct location and assertion of location is important.
5.2.4 Confidentiality of process-related metadata	This is important, but no specific provisions are noted.
5.2.8 Secure clean-up	Some functions are likely to be defending against virus, or other attacks, or will be observing cryptographic or user service data. Therefore secure clean up is important.
5.2.10 Assurance of compliance by hosting service	Given that these functions control access to other VNFs and are instrumental in defending the network edge, it needs to be possible to assert they are operating correctly.
5.2.11 Availability of entropy source	
5.3.1 Secure routing/switching	These functions may include network switches, routers or other secure routing/switching functionality.
5.3.2 Workload placement policy and operation security	These functions are network security functions.
5.3.3 Availability of an attestation authority	These functions are network security functions.

4.4.3 Notes on the technologies and measures in the context of Network Monitoring & Control Functions

The following notes give specific interpretation of clause 6 in the context of Network Monitoring & Control Functions.

Table 4

Clauses of the present document	Notes for Network Monitoring & Control Functions
6.1 Memory inspection	Memory inspection by a hosting service may cause issues for some functions. However memory inspection may be a tool that some functions require to be able to monitor or control other VNFs.
6.2 Secure logging	Secure logging especially for post event analysis is important.
6.3 OS-level access control and 6.4 Post-incident analysis	This is important, but no specific provisions are noted.
6.5 Physical controls and alarms and 6.6 Personnel controls and checks	This is important and will depend on the specific function, but no specific provisions are noted.
6.8 Read-only partitions and 6.9 Write-only partitions	Read or write-only partitions may be required by functions.
6.11 Communications Security	These functions are network security functions.
6.12 Measured boot and 6.13 Secured boot	This is important, but no specific provisions are noted.
6.14 Constant resource usage	Unlikely to be necessary.
6.15 Attestation, 6.16 Hardware-mediated execution enclaves and 6.17 Trusted Platform Module (TPM)	Hardware-mediated execution enclaves/TPM etc. TPMs or equivalent implementations that include the protection capabilities, as provided by TPMs, may be mandated to be hardware based.
6.18 Self-encrypting drives/storage	Unlikely to be necessary.
6.19 Direct Memory Access to hardware resources	May be required in order for the function to perform monitoring and control of other VNFs.
6.20 Hardware Security Modules	Some use cases may require access to the security capabilities offered by HSMs.
6.21 Software integrity protection and verification	Software integrity protection and verification is expected to be required for almost all use cases.

4.5 Lawful Interception

4.5.1 Introduction and baseline references

This case covers the underlying requirement for Lawful Interception. The baseline requirements for Lawful Interception are listed in ETSI TR 103 331 [i.3] (with specific handover requirements covered in ETSI TS 102 232 [i.4]) and ETSI GS NFV-SEC 004 [i.12] provides an analysis of applying the requirements from ETSI TS 101 331 [i.5] in the NFV.

4.5.2 Applicability of security requirements in the context of Lawful Interception

The following notes give specific interpretation of clause 5 in the context of LI.

Table 5

Clause of the present document	Notes for LI
5.0.1 Overview	Availability has non-intuitive but important implications for LI functionality. Regarding "it may be preferable to *lose* availability if the only alternative would be drop to a less secure system or mechanism": prioritization of LI should be equivalent to the prioritization of the underlying service.
5.0.2 Prevention versus remediation	Remediation rather than prevention: Rapid monitoring and reporting of loss of LI services would enable timely remediation if functionality was unavailable.
5.1.1 Capability assertion and attestation at boot-time to 5.1.5 Assert continued integrity of own system at run-time	These are important to LI, but no LI-specific provisions are noted.
5.1.6 Location assertion	Location assertion. This is critical to LI functionality. An essential requirement is that the LI takes place in the country that issued the authorization (this means that any LI-VMs plus the network function VMs that are being monitoring are mandated to be in-country).
5.2.1 Confidentiality of data	Confidentiality of data: The target list is critical information, which should be protected such that only the appropriate LI functions are entitled to read or modify this information. The product of interception is also mandated to remain confidential on the delivery route to the LEA.
5.2.2 Confidentiality of data-related metadata to 5.2.4 Confidentiality of process-related metadata	Confidentiality of meta-data, processes, related meta-data. Critical to LI but no specific provisions are noted.
5.2.5 Concealment of resource usage	These are important to LI, to reduce opportunities for detection of when LI is taking place. The critical requirement is to minimize any changes that are visible to non-authorized parties when LI starts or stops. There is also a requirement for overall statistics about LI to be difficult to determine.
5.2.6 Secure communications	It will be important to maintain secure communications for LI administration (i.e. with LI management functionality) and for interception product delivery (with mediation and handover functions). The location of these functions is for further study.
5.2.7 Secure storage	Secure storage of certain information is critical: secure storage of target list is important. However, the volume of persistent LI-related secure storage is typically small for many use cases, but can be high for some data types.
5.2.10 Assurance of compliance by hosting service	Assurance of compliance by hosting service. From an LI point of view: trust in hosting service needs to come from external entity, which could be an HBRT (e.g. TPM) within the NFV platform or could be the ADMF if this is external.
5.3.2 Workload placement policy and operation security	Workload placement policy will be critical for LI. Migration of services has particular implications: it would be important to ensure that target services are not migrated to points where it would be impossible or improper (e.g. into other jurisdictions) for the related LI functionality to follow.

4.5.3 Notes on the technologies and measures in the context of Lawful Interception

The following notes give specific interpretation of clause 6 in the context of LI.

Table 6

Clause of the present document	**Notes for LI**
6.1 Memory inspection	Memory inspection by a hypervisor can cause non-authorized disclosure issues for LI. In general hypervisors are not trusted to introspect LI details on hosted services.
6.2 Secure logging	There is a requirement for logging for LI purposes but this is mandated to be in LI-specific secure logs which are protected from other secure storage. Unless there is a specific reason, the details that are logged should be less sensitive information (e.g. LI reference numbers, or dates and times that target lists were updated) rather than more highly sensitive information (e.g. target details). For application details - refer to appropriate application standards (e.g. 3GPP).
6.3 OS-level access control and 6.4 Post-incident analysis	This is important to LI, but no LI-specific provisions are noted.
6.5 Physical controls and alarms and 6.6 Personnel controls and checks	There exists considerable experience in managing these aspects for LI capability. Typically they are defined on a national level but in general there is a considerable amount of requirements in common.
6.8 Read-only partitions and 6.9 Write-only partitions	Read or write-only partitions. No specific LI requirements noted.
6.11 Communications Security	Confidentiality and integrity are critical. The quality of service requirements is typically comparable to those for the underlying service. **Specifically for LI delivery:** Requirements for LI delivery require headroom above the bandwidth of the target service to support LI meta-data and LI delivery headers LI is intended to be delivered in real-time with minimal latency. **LI triggering:** Triggering functions should happen quickly enough to ensure that all authorized information can be captured (i.e. information from signalling functionality needed to trigger the start of content interception).
6.12 Measured boot, 6.13 Secured boot	Important but no LI-specific provisions are noted.
6.14 Constant resource usage	Constant resource usage. As per the text in clause 6.14, constant resource usage is not per se a requirement, but approaches to hiding changes in resource usage is mandated to be resistant to statistical discovery attacks.
6.15 Attestation, 6.16 Hardware-mediated execution enclaves and 6.17 Trusted Platform Module (TPM)	TPMs or equivalent implementations that include the protection capabilities, as provided by TPMs, may be mandated to be hardware based. Virtual modules may not be sufficiently robust. FOR FURTHER STUDY: These are likely to play an important role in LI provision and further details are required.
6.20 Hardware Security Modules	FOR FUTHER STUDY: These may be required by LI components.
6.21 Software integrity protection and verification	Software integrity protection and verification is expected to be required for almost all use cases.

4.6 Retained Data

4.6.1 Introduction and baseline references

This Use Case covers the underlying requirement for Retained Data. The baseline requirements are listed in ETSI TS 102 656 [i.6], with specific handover requirements covered in ETSI TS 102 657 [i.7]. A more detailed treatment of Retained Data considerations in an NFV environment is given in ETSI GS NFV-SEC 010 [i.18].

For this clause of the present document, "Data" refers to any data which is subject to national Retained Data regulations. The present document is not a legal document and makes no implication of when or whether Data should or should not be retained or disclosed.

This Use Case considers the following stages of the Retained Data process, looking solely at the aspects which are specific to NFV:

- the collection of Data
- the storage of Data
- the querying mechanism
- the delivery of requests and the handover of results

For example, this Use Case includes the situation where Retained Data queries are being handled by a network component which is virtualised and the RD storage is not necessarily in the same jurisdiction as the users of the service or the agency which is making the request. In this example, security requirements are critical to ensure that appropriate access controls are enforced, that appropriate privacy requirements are met and that the confidentiality of the list of subjects of interest is maintained.

4.6.2 Applicability of security requirements in the context of RD Storage and Query

The following notes give specific interpretation of clause 5 in the context of RD Storage and Query.

Table 7

Clause of the present document	Notes for RD Storage and Query
5.0.1	Availability is important to RD functionality. Short outages can be tolerated, provided there is clear information about when the outage is happening, and that it can be recovered promptly, and the any backlog can be cleared quickly.
5.0.2	Remediation rather than prevention is acceptable for RD availability: rapid monitoring and reporting of loss of RD services would provide some remediation if functionality was unavailable, provided mechanisms were in place to restore the functionality.
5.1.6	Location assertion. This is important for RD storage. It is important to know where the RD storage is taking place, to gain an understanding of security or legal concerns.
5.2.1	Confidentiality of data: The list of subjects of interest is important to keep confidential, both for operational reasons and also to protect the privacy of the subject of interest.
5.2.8	Secure communications. It will be important to maintain secure communications for requests and for responses.
5.2.9	While a request is being answered, it is important that the details of that request are stored securely. Once the query has been answered, it is important to follow national regulations and requirements regarding when the request and response can be deleted. In general there is no need to store the sensitive details in the request or response, provided: - Sufficient measures are in place to be able to provide evidential assurance about the material if it is later used in court (e.g. through a regime based on a hash or digitally signature). - From an audit point of view, it may be necessary to store certain information about the request i.e. the time, and a unique ID for the request.
5.2.11	Assurance of compliance by hosting service. From an RD point of view: trust in hosting service needs to come from external entity, which could be an HBRT (e.g. TPM) within the NFV platform or could be an RD management function if this is external.
5.3.2	Workload placement policy will be important for RD. Migration of services is difficult: it would be important to ensure that target services are not migrated to points where it would be impossible for the related RD functionality to follow.

4.6.3 Notes on the technologies and measures in the context of RD Storage and Query

The following notes give specific interpretation of clause 6 in the context of RD Storage and Query.

Table 8

Clause of the present document	Notes for RD
6.1	Memory inspection by a hypervisor would cause issues for RD storage. In general, hypervisors are not trusted to inspect RD requests on hosted services.
6.2	In order to provide evidential assurance and audit, RD storage and query functions are required to have logging features. Unless there is a specific reason, the details that are logged should be non-sensitive information (e.g. RD reference numbers, dates, time of queries and hashes or signatures if appropriate) rather than sensitive information (e.g. personal details).
6.5 and 6.6	Physical and personnel controls. There is considerable existing experience in managing these aspects for RD capability. Typically they are defined on a national level but in general there is a considerable amount of requirements in common. In general these requirements are likely not to be NFV-specific and therefore this is not handled further here.
6.11	Communications security. Confidentiality and integrity are critical. The response time is not real-time but longer delays (e.g. 1 minute) are not acceptable in threat-to-life situations.
6.14	No requirement for constant resource usage.
6.15, 6.16 and 6.17	Hardware-mediated execution enclaves/TPM are likely to be important in order to meet national RD requirements. TPMs or equivalent implementations that include the protection capabilities, as provided by TPMs, may be mandated to be hardware based.
6.20 Hardware Security Modules	Some RD functions may require access to the security capabilities offered by HSMs.
6.21 Software integrity protection and verification	Software integrity protection and verification is expected to be required for almost all use cases.

4.7 Personally Identifiable Information protection

4.7.1 Introduction

The protection of Personally Identifiable Information (PII) associated with customers is an important regulatory driver in many locations. Guidelines and regulations related to PII protection widely between different jurisdictions, for example the United States, NIST 800-122 [i.9] and European Union member states, Directive 95/46/EC [i.10]. Likewise, the specific definition of what information needs to be protected, and even the terms used (in some jurisdictions, "personal information" is preferred, for example) also vary. As a result, clear guidance on this issue is beyond the scope of the present document. However, PII exists in the context of many hosted applications, and its confidentiality and integrity are both important to protect. The next clause addresses security requirements in the context of PII. Given the differing regulations and best practice guidelines across jurisdictions, no discussion of specific measures is provided.

4.7.2 Applicability of security requirements in the context of PII protection

The specific requirements for PII protection will vary, but the following are considered key:

- 5.1.1 Capability assertion and attestation at boot-time.
- 5.1.2 Capability assertion and attestation at run-time.
- 5.1.3 Assert secure provision of hosted application.
- 5.1.4 Assert own system integrity at boot.
- 5.1.5 Assert continued integrity of own system at run-time.
- 5.1.6 Location assertion.

- 5.2.1 Confidentiality of data.
- 5.2.7 Secure storage.
- 5.2.8 Secure clean-up.
- 5.2.10 Assurance of compliance by hosting service.
- 5.3.2 Workload placement policy and operation security.
- 5.3.3 Availability of an attestation authority.

The requirements on the hosting system are due to the need for certainty that the hosted application is being hosted on an appropriate platform, and are tied to the workload placement policy and operation security (clause 5.3.2) and location assertion (clause 5.1.6), which is also important where services are being provided across jurisdictional boundaries. Not all hosted applications will require secure storage (clause 5.2.7), but confidentiality of data (clause 5.2.1) and secure clean-up (clause 5.2.8) at the appropriate point in the hosted application's lifecycle are the key requirements for this use case.

5 Security requirements

5.0 Void

5.0.1 Overview

There are, to follow the classic "CIA" model of security provision, three properties that a hosted application may expect from its hosting service: Confidentiality, Integrity and Availability. Although availability is an important property, from the point of view of multi-layer administration, it is the first - confidentiality - and the second - integrity - which are of most relevance. Confidentiality is obviously important, as without confidentiality, a hosted application can have no certainty that its actions, its authentication tokens, cryptographic keys, sensitive algorithms, etc., are not subject to being exposed to the hosting service. Integrity may be a less obviously necessary property, but if the hosting service can change the data or processes within a hosted application, then damage can still be done. Many cryptographic suites and techniques guarantee confidentiality, and if not guaranteeing integrity, provide built-in measures by which a loss of integrity can be quickly detected. In addition, the attestation of confidentiality and availability will need to show clearly the levels attained, as well as any violations.

Guaranteed availability is less easy to provide, particularly because controls over resource usage are generally under the control of the hosting service. External monitoring measures applicable to all of the identified use cases would allow steps to be taken externally to the hosting service to remediate any loss of availability. The timeliness of any detection will vary from use case to use case, and for some cases, it may be preferable to *lose* availability if the only alternative would be drop to a less secure system or mechanism. Availability is therefore considered out of scope of the present document except where particularly identified.

There are a number of types of attacks or security failures that may occur. Availability attacks, including resource starving and isolation attacks, are not in the scope of the present document. Others may be simply categorized thus:

- **Active hostile:** this category includes attempts by the hosting service to impact on the capability of the hosted application to perform as expected. A number of availability attacks would fall under this category, but attacks on the integrity of the service are also relevant. An active attack involves the attacker interacting in some manner with the target application by one or more of the following techniques:
  - through its normal operation;
  - by injecting known or probable pathological inputs;
  - by manipulating its environment.

Both the application itself and the NFVI hold roles in mitigating these types of attacks:

- **Passive hostile:** this category includes attempts by the hosting service to gain unauthorized access to data or processes being run by the hosted application, or to gain information about those data or processes. A passive attack involves the attacker using only observed inputs, outputs or memory of the target application or side effects of its operation, but without any direct interaction with it (e.g. adding/changing any inputs or blocking/manipulating any outputs).

The NFVI has a primary, perhaps singular, role in detecting and eliminating such attacks because, by definition, they cannot be mitigated by the application itself:

- **Accidental:** this category includes the possibility that information may leak or be discovered by unauthorized parties who are not specifically looking for information from the hosted application. Where human, they might typically be administrators of the hosting service or have privileged access to components within or without the hosting service. Other accidental failures might arise from insecure logging techniques, file descriptor leakage or router/switch misconfiguration.

It should be noted that not all attacks are directly related to the hosting service itself, as there may exist side channel attacks and vulnerabilities in the supporting systems. Where possible, these should be considered, and mitigations put in place to reduce impact. They are not considered explicitly within the scope of the present document.

Knowing that there is activity in the hosted application - that resources are being consumed in a particular manner or matching a particular usage template - or detecting changes in resource usage, over various time scales, may be enough for some attacks to be considered "successful".

5.0.2 Prevention versus remediation

For some use cases, the amount of effort expended to prevent failures in meeting requirements through various techniques and assurances may outweigh the benefit of meeting the requirements. Equally, balancing the level of preventative measures with rapid monitoring and reporting to allow quick investigation and rapid remedial action allows sensible and appropriate resources to be applied most efficiently.

5.0.3 Channels for assertions by the hosting service

A number of the requirements on the hosting service mean that it makes assertions as to its capabilities and state to other parties - typically other components in the NFV deployment. Of the three security properties noted above, availability and integrity of data are typically more important than confidentiality - in this case, the priorities are different to the usual - and although information about the ability (or inability) of a hosting service to provide a particular capability or set of capabilities may be of interest to an attacker, the components choosing to site hosted applications on hosts should be taking such information into consideration before selecting a particular hosting service.

It should be noted, also, that hosting services will generally expect - and will need to enforce that - only authorized parties are making requests for such information (for the reasons noted above), so standard techniques should be utilized to ensure that identities are checked. The obvious approach is to encrypt the communication channels and to use cipher suites which support authentication to provide identities which can thence be used to make decisions about authorization.

The need for the requesting parties to validate the identity of the hosting service is much greater, as a hosting service which is incorrectly identified could lead to hosted applications being sited on inappropriate hosting services, with concomitant reductions in security. This is considered a possible attack, and since a VM (such as a compromised VNFCI) on a trusted hosting service could masquerade as a hosting service, various techniques including authentication and secured routing and switching (see clauses 5.3.1 and 5.2.9) should be employed for these communication channels.

5.0.4 The value of assertions

While a hosting service may make assertions across one more of the identified requirements, the value attached to these assertions is purely to the hosted application or associated management functions, and may be a function not only of the trust placed in the hosting service, the broader NFVI and its associated management and orchestration components, but also in non-technical considerations such as Service Level Agreements, legal frameworks, contractual obligations and human relationships.

A hosted application or associated management functions may doubt the correctness or value of one or more of the assertions made by the hosting service - or related functions - and choose to act, or not act, on that doubt. Typically, such doubts will be used as an input into broader trust maintenance considerations and may decrease the level of trust held with respect to *other* assertions by the hosting service and associated functions.

5.0.5 Use cases to requirements mapping

Table 9

	4.1 Multi-tenant hosting	4.2 Infrastructure as a service (IaaS)	4.3 Security Sensitive Application Functions	4.4 Security Network Monitoring & Control Functions	4.5 Lawful Interception	4.6 Retained Data	4.7 Personally Identifiable Information Protection
5.1.1 Capability assertion and attestation at boot-time	X	X	X	X	X	X	X
5.1.2 Capability assertion and attestation at run-time			X	X	X	X	X
5.1.3 Assert secure provision of hosted application	X	X	X	X	X	X	X
5.1.4 Assert own system integrity at boot	X	X	X	X	X	X	X
5.1.5 Assert continued integrity of own system at run-time			X	X	X	X	X
5.1.6 Location assertion	X	X	X	X	X	X	X
5.2.1 Confidentiality of data	X	X	X		X	X	X
5.2.2 Confidentiality of data-related metadata			X		X	X	
5.2.3 Confidentiality of processes	X	X	X		X		
5.2.4 Confidentiality of process-related metadata			X	X	X		
5.2.5 Concealment of resource usage			X		X		
5.2.6 Secure communications	X	X	X		X	X	
5.2.7 Secure storage	X	X	X		X	X	X
5.2.8 Secure clean-up	X	X	X	X	X	X	X
5.2.9 Secure routing/switching			X		X		
5.2.10 Assurance of compliance by hosting service	X	X	X	X	X	X	X

	4.1 Multi-tenant hosting	4.2 Infrastructure as a service (IaaS)	4.3 Security Sensitive Application Functions	4.4 Security Network Monitoring & Control Functions	4.5 Lawful Interception	4.6 Retained Data	4.7 Personally Identifiable Information Protection
5.2.11 Availability of entropy source	X	X	X		X		
5.3.1 Secure routing/switching			X	X	X		
5.3.2 Workload placement policy and operation security	X	X	X	X	X	X	X
5.3.3 Availability of an attestation authority	X	X	X	X	X	X	X
NOTE: X = required.							

5.1 Requirements - hosting service

5.1.1 Capability assertion and attestation at boot-time

The hosting service will need to be able to make assertions about the various capabilities that it can offer, and also be able to attest these to a third party. These capabilities may be separated into the following categories.

Table 10

	Boot-time only attestable	**Run-time attestable**
Static capabilities	Capabilities which are not expected to change during normal operation, but which are only attestable at boot-time.	Capabilities which are not expected to change during normal operation, but which may be attested at run-time (to check for malfunctions, etc.).
Dynamic capabilities	Capabilities which may change during normal operation, but which are only attestable at boot-time.	Capabilities which may change during normal operation, and which may be attested at run-time.

As boot-time attestation takes place, by definition, before the hosting service starts to host the hosted application, communication of the attested capabilities will need to be to a party (an "attestation authority") with which the hosted application, once it is provisioned, will have a two-way trust relationship. There may not seem any specific need for the hosting service to have a trust relationship in the direction of the attestation authority, but the very act of attesting itself to the attestation authority sets up a trust relationship in the other direction. As such trust, relationships may in fact be very important. In certain use cases, there may be multiple attestation authorities, in different trust domains. To give three examples:

- The main NFV deployment's attestation authority, under the control of the hosting operator, and part of their management and orchestration trust domain.
- An attestation authority within the trust domain of an LEA for use with Retained Data or Lawful Interception capabilities.
- An attestation authority providing information to the owners or operators of third party hosted services, such as in the multi-vendor use case.

Capabilities may include any of the requirements laid out in this clause, though the description or level of granularity may be different to the descriptions provided in the present document.

5.1.2 Capability assertion and attestation at run-time

Note the table 10 in clause 5.1.1, which categorizes the different types of capabilities based on the static or dynamic nature and on when they may be attestable.

There are three types of occasion on which a party may wish to have an attestation of the hosting service's capabilities at run-time:

- General health-check: a check on the current health of the hosting service, typically carried out periodically.
- Suspected problem: when a problem, error, failure or attack is suspected.
- Change of use: when capabilities which have not previously been used are required, and a check is required on their availability and health.

The exact mechanism by which an attestation takes place may be of various different types, and may be periodic (a "push" from the hosting service similarly to a heartbeat) or triggered by a request from another entity, including the hosting application. However, it is generally expected that the communication will take place via the "attestation authority" identified in clause 5.1.1 for several reasons:

- Maintaining a separate record of attestations for boot-time and run-time would be wasteful and could cause inconsistencies.
- The hosted application is generally not expected to have sufficient intelligence to be able to make a direct decision about the level of trust it should place in the hosting service see clause 5.0.4, and this should therefore be delegated to a third party.
- The time when an hosted application's (delegated) decision to trust the hosting application takes place may not coincide with the availability of an attestation event, and so stored information about attestations may need to made available.

Capabilities may include any of the requirements laid out in this clause, though the description or level of granularity may be different to the descriptions provided in the present document.

5.1.3 Assert secure provision of hosted application

In order for an application - and any remote management functions - to be assured that it can operate correctly, it will first need to be assured that it was securely provisioned. This could be through instantiation of a new instance, resumption of an existing instance, or migration of an instance to the asserting hosting service. To satisfy this requirement, the hosting application will need to assert that the one or two separate conditions have been met:

- integrity of provision: that the application has been provisioned as packaged, with no changes by in transit or by the hosted application, to the best knowledge of the hosting service. Note that integrity of the application in storage is beyond the scope of control of the hosting service.
- confidentiality of provision: that any data or processes (and associated metadata) marked as confidential has not, to the best knowledge of the hosting service, been exposed to unauthorized parties or components (see clauses 5.2.1, 5.2.2, 5.2.3 and 5.2.4).

5.1.4 Assert own system integrity at boot

In order for an application - and any remote management functions - to be assured that the hosting service on which it is executing should be trusted to host it correctly and to make other assertions about its operation, the hosting service will need to be able to assert its own system integrity at boot time.

5.1.5 Assert continued integrity of own system at run-time

In order for an application - and any remote management functions - to be assured that the hosting service on which it is executing should be trusted to host it correctly and to make other assertions about its operation, the hosting service will need to be able to assert its own continued system integrity at run time Such assertions may be made either on a cadence under the control of the hosting application or at the request of other parties, including the hosted application and associated management functions.

5.1.6 Location assertion

Some applications may require that they operate only in a particular geographical, logical or jurisdictional location or set of locations. Determining this with high assurance is essential. In order to meet this requirement, a hosting service will need to be able to assert its location across one or more of these parameters.

5.2 Requirements - hosted application

5.2.1 Confidentiality of data

A hosted application may have specific sets of data which are required to be kept confidential from other applications or from the hosting service. The hosted application needs to be able to specify what set(s) of data should remain confidential, and the parties authorized to access it (a whitelist) or the parties not authorized to access it (a blacklist).

5.2.2 Confidentiality of data-related metadata

In certain use cases, knowledge about the amount or type of data may give away information which is of use to an attacker. A hosted application may have specific sets of data for which it requires the related metadata to be kept confidential from other applications or from the hosting service. The hosted application needs to be able to specify what set(s) of metadata should remain confidential, and the parties authorized to access it (a whitelist) or the parties not authorized to access it (a blacklist).

5.2.3 Confidentiality of processes

A hosted application may have specific sets of processes which it requires to be kept confidential from other applications or from the hosting service. The hosted application needs to be able to specify what set(s) of processes should remain confidential, and the parties authorized to access it (a whitelist) or the parties not authorized to access it (a blacklist).

5.2.4 Confidentiality of process-related metadata

In certain use cases, knowledge how many processes of a particular type are executing may give away information which is of use to an attacker. A hosted application may have specific sets of processes for which it requires the related metadata to be kept confidential from other applications and/or from the hosting service. The hosted application needs to be able to specify what set(s) of metadata should remain confidential, and the parties authorized to access it (a whitelist) or the parties not authorized to access it (a blacklist).

5.2.5 Concealment of resource usage

For some use cases, information about the hosting application's resource usage may be of interest to an attacker. Various types of resource usage may be relevant, including:

- RAM.
- CPU.
- Storage.
- Network I/O.
- Cache usage.
- Hardware offload usage (including crypto offload).

The hosting application may therefore wish to conceal the usage of one or more of these types of resource from other parties, including other applications and the hosting service.

Various techniques may be utilized, some of which may require assistance from the hosting service. Note also the points around guaranteed switching and routing made in clause 5.2.9. It should also be noted that some of the available techniques may have an impact on power use, heat management and equipment life.

5.2.6 Secure communications

A hosting application may require the ability to communicate securely with other parties, which may be remote or on the same hosting service. There may be a requirement to secure these communications from:

- The hosting service.
- Other applications on the hosting service.
- Other parties in the NFVI domain.
- Parties within the Management and Orchestration domain.
- Parties beyond the NFV deployment.

There are three separate life-cycle states for secure communications:

- Provisioning: may require access to key information, cipher suites, PKI elements and network connection.
- Maintenance: since key re-establishment may be required periodically, this may require access to the same elements as the provisioning state.
- Tear-down (de-provisioning): may require assurances from the hosting service that the network connection has been correctly terminated, see clause 5.2.8.

A pre-requisite for secure communications may be the availability of an appropriate entropy source to allow encryption to be adequately provisioned: see clause 5.2.11 for more details.

5.2.7 Secure storage

Secure storage may be required to keep data confidential over a period of time. Availability and integrity of this data are both expected to be key considerations. Various types of storage may be offered by the hosting service, and though the operation of the various types of storage may be transparent to the hosted application, certain properties may be important. At provisioning time, therefore, the hosting application may request specific properties, and the hosting service may need to be able to characterize the various types of storage available.

Types of storage include:

- RAM.
- Non-volatile RAM.
- Solid-state drive.
- Hard drive.
- Network-attached storage.
- Block storage.
- Object storage.

Properties of storage include:

- Journaled.
- Low-latency.
- Transactional.

- RAID.
- Local.
- Remote (in which case location may be important for some use cases).
- Key management options.

Note that there are various stages in the lifecycle for secure storage, including:

- *Provisioning*: transport and/or storage keys should be provisioned; policies for key life-cycle, back-up, etc. put in place.
- *Maintenance*: maintaining availability, managing key life-cycle, managing back-up, failure and recovery.
- *De-provisioning*: secure clean-up, as appropriate, see clause 5.2.8.

In some (extreme) cases, the use of wrote-only storage may be required, see clause 6.9.

5.2.8 Secure clean-up

There is a number of hosting application lifecycle events which may require secure clean-up of resources. In some cases, the hosting application may be able to perform this secure clean-up on its own or with related management functions. In other cases, it may require the assistance of the hosting service, other NFVI components or components in the management and orchestration domain. For more information, see ETSI GS NFV-SEC 003 [i.2].

5.2.9 Secure routing/switching

Where the hosting service also provides routing or switching services to the hosted application (which may not always be the case - see, for example, clause 6.19), this should be subject to the same levels of availability as those governed by the Service Level Agreements of the standard service traffic, *or higher*. Specifically, in the example of a Lawful Interception service, the hosted LI application needs to be able to capture 100 % of the target service. The hosted LI application in this case will also need to be able to transfer all captured packets to its LEA mediation function with 0 % subsequent loss.

See also clause 5.3.1.

5.2.10 Assurance of compliance by hosting service

In order for the hosted application to be assured that it can trust that its particular requirements are being met by the hosting service - and, where applicable, other components - it will need to be assured that the hosting service is compliant. In order to do this, a third party will generally need to provide such services - an "attestation authority", see clause 5.3.3. There is a "turtles problem" here, in that the hosted application will need to have some trust in the hosting service, as otherwise it cannot run in order to communicate with the attestation authority. If, once it has communicated with the attestation authority, sufficient assurances of compliance are not forthcoming, the hosted application may choose to change its behaviour, and/or any external parties may choose not to trust it.

See ETSI GS NFV-SEC 003 [i.2] and clause 5.0.5 for more details.

5.2.11 Availability of entropy source

The generation of secure keys for many cryptographic operations requires a certain degree of randomness - the exact amount being specific to the type of key, the operation, and the level of security required. Entropy is the randomness which is provided to processes requiring it, and generally requires a hardware source.

There are three specific requirements on entropy sources:

- Trustedness - it will not be subject to tampering by another party, which may include the hosting service.
- Randomness - it will provide appropriate levels of randomness as required by the hosted application.
- Availability - it will provide the required random numbers in a timely manner and in sufficient quantity.

Provision of appropriately random entropy with high availability is a notoriously difficult task, and many approaches have been shown to be flawed. State of the art generally requires at least one hardware source as an input, and various implementations are available. Approaches such as sampling network traffic are not considered acceptable. Industry best practice should always be followed.

5.3 Requirements - other components

5.3.0 Introduction

This clause records requirements which may be sited outside of the scope of the hosting service and hosted application, or where part of the functionality may do so. As such, they may be considered out of scope, but given their importance, they are listed here for completeness.

5.3.1 Secure routing/switching

The routing/switching fabric which may have impact on the hosting service and hosted applications embraces all parts of the NFVI, and, by the broadest definition, some entities which sit outside it, either because they are part of legacy deployments, or because they are external to the NFV deployment itself. It may also include any controller entities such as an SDN controller. The fabric may include, for instance:

- SDN Controllers.
- Top of Rack (ToR) switches.
- Routers in the network.
- Physical switches in the network.
- Physical switches with software elements (e.g. stand-alone or SDN-controlled "soft switches").
- Virtual switches in an NFVI host (including, in the nomenclature of the present document, as part of a hosting service).

The key requirement for security within the routing and switching fabric is for availability. Given the requirements for confidentiality of all communications see clause 5.2.6, it is expected that all sensitive traffic will be encrypted, and that appropriate cipher suites will be used to detect possible failures in integrity in the transmitted data. It is the availability of the data, then, in a timely manner, which is the specific requirement. Routes used for transmission of data from hosting applications requiring secure routing or switching should be subject to the same levels of availability as those governed by the Service Level Agreements of the standard service traffic, *or higher*. Specifically, in the example of a Lawful Interception service, a hosted LI application needs to be able to capture 100 % of all target service. The hosted application in this case will also need to be able to transfer all products of interception to the LEA mediation function with 0 % subsequent loss.

5.3.2 Workload placement policy and operation security

Hosted applications that require multi-layer administration capabilities should only be allowed to execute (whether through initial instantiation or migration) on hosting services which provide the necessary capabilities to meet the requirements of the particular use case. Typically, this will require some or all of the following:

- Attestation of the available hosting services at provisioning time.
- Attestation of the available hosting services at boot-time.
- Attestation of the available hosting services at run-time.
- Selection of a set of hosting services to meet the requirements of particular use cases on specific hosted applications.
- Policies to ensure that instantiation or migration of relevant hosted applications only occurs on or to compliant hosting services.

- Detection of changes in compliance status for hosting services.
- Policies to ensure halting of execution of hosted applications on hosting services which move to a non-compliant state, and any clean-up activities associated with this lifecycle stage.

All of these issues lay outside the scope of the hosting service and are expected to reside, instead, in the VIM or the VNFM. Of particular concern are the integrity of the policies and importance of ensuring that their operation is correctly carried out: attacks impacting on either of these might easily have the effect of leaving the security of the overall system significantly compromised.

Levels of trust in the Management and Orchestration components are outside the scope of the present document, but are also key to the overall system and its operation. This trust exists in at least two contexts:

1) That the Management and Orchestration components are not subject to tampering by unauthorized parties, either in their configuration or operation.

2) That the operation of the Management and Orchestration components is dependable and predictable.

NOTE: However, the importance and significance of clause 5.2.10.

5.3.3 Availability of an attestation authority

In order for assurances around appropriate levels of trust to be made in the system as a whole, a number of requirements are dependent on the availability of an attestation authority. This may be part of the more general Management and Operations trust domain, or exist within a separate trust domain specific to the hosted application.

See also clauses 5.1.1, 5.1.2 and 5.2.10.

The key reference for this issue should be considered to be ETSI GR NFV-SEC 007 [i.8].

6 Available technologies and measures

6.0 Introduction

This clause provides a list of technologies and measures from various domains. The intent is to build a list of technologies and measures which can be combined using different approaches and architectures clause 7 to meet the requirements of the various use cases, see clause 4.

6.1 Memory inspection

6.1.0 Introduction

Hosting service

Memory inspection is the ability of one process to "peek" into the memory assigned to another process, and can be granted by one entity to another. This generally refers specifically to the ability to look into the memory assigned to a particular virtual machine or Container. By default, all of the well-known and widely-deployed hypervisors (e.g. KVM, Xen, ESXi, Hyper-V) have the ability to inspect the memory of virtual machines on the same host, as well as write to it. They also have the ability to write into such memory - though this is not exactly "inspection". They also have the ability to "grant" access for applications or VMs to read and/or write sections of memory owned by other applications or VMs.

Linux-based Container systems have a similar property, in that the hosting Linux Operating System processes are able to inspect the memory assigned to Containers on the same host.

A system which enjoys full memory inspection capabilities should also be expected to enjoy the ability to see process-related data within the hosted application.

Memory inspection can be used as an attack vector or as a security enabler.

6.1.1 Memory inspection as an attack vector

Since this clause describes an attack vector, it is not entirely appropriate to list is as an available technology or measure. However, since it lies at the bottom of many of the possible attacks on systems and makes the application of many of the requirements listed in clause 5 technically challenging to meet, a description is provided here.

The key capability of the hosting system's administrative function to be able to inspect any portion of memory which is physically attached to it, and its ability to grant this capability to other applications or processes, means that no hosted application running on the system can be sure of the confidentiality or integrity of its executable code or data. This relates:

- directly to the requirements laid out in clauses:
  - 5.2.1 Confidentiality of data;
  - 5.2.2 Confidentiality of data-related metadata;
  - 5.2.3 Confidentiality of processes;
  - 5.2.4 Confidentiality of process-related metadata;
- and indirectly to requirements laid out in clauses:
  - 5.2.5 Concealment of resource usage;
  - 5.2.6 Secure communications;
  - 5.2.7 Secure storage;
  - 5.2.9 Secure routing/switching;
  - 5.2.11 Availability of entropy source.

Other requirements may also be affected.

Note that users with sufficient administrative control over the hosting service can also claim this capability or grant it to another process.

6.1.2 Memory inspection as a security enabler

Memory inspection can also be as a capability for hosting applications. In the Lawful Interception use case, for example, one approach for the hosted application (the LI VM or Container) is for the hosting service to grant it rights to perform memory inspection on one or more service applications. In this case, memory inspection is a technology which can be used to meet requirements of a hosted application.

6.2 Secure logging

Hosting service

Secure logging may encompass at least three properties:

1) Creation of entries which are confidential from other parties. This may be by encrypted communication with a party to whom access is restricted or by creation of encrypted entries.
2) Creation of entries whose integrity can be checked at the entry level.
3) Creation of a chain of entries, where the chain itself can be checked for tampering or deletion.

6.3 OS-level access control

Hosting service

OS-level access controls such as SELinux and AppArmor® (see note) provide mechanisms for supporting access control security policies on Operating System processes. For example, access control policies can be applied to processes attempting access to files and network resources, and minimum privilege applied to reduce the damage that could be caused even by an authorized process, should it be compromised.

NOTE: AppArmor® is the trade name of an open source software from SUSE® LLC. This information is given for the convenience of users of the present document and does not constitute an endorsement by ETSI of the product named. Equivalent products may be used if they can be shown to lead to the same results.

Creation of policies for application of OS-level access controls can be complex, and changes hosted application's use of Operating System capabilities require revision of the policies, so there is an operational overhead associated with their usage. However, the amount granularity that such approaches provide can be considerable, and careful policy application, combined with log auditing and real-time controls, may provide significant security benefits.

6.4 Post-incident analysis

Hosting service, hosted application, other systems

Post-incident analysis is the checking of various logged measurements to establish details of the attack, i.e. the mode and method of attack, the time of the attack, the identities or locations of attackers. Typically this is for forensic purposes, but may be for other reasons such as counter-attacks.

Measurements typically checked include:

- Applications and traffic logs, both from the hosting service itself and from other systems.
- Integrity of physical controls.
- Integrity of software, firmware and BIOS.
- Modifications of application data or permissions.
- Modifications of hosting service data or permissions.
- File integrity analysis.
- Human reports of anomalous system behaviour.
- MAC time analysis.
- Searches for malware.

It is now understood that very sophisticated attacks never end. Therefore, the "post"-incident analysis should be supplemented by further (and ongoing) analysis of current traffic patterns, anomalous application behaviour, virus scans, etc.

A deeper discussion of this topic is out of scope for the present document, but many guides to industry best practice are available, including a collection on forensic analysis provided by the SANS organization [i.13].

6.5 Physical controls and alarms

Hosting service, other systems

Physical controls and alarms are all those measures which are based not in software, but in hardware. Generally, these refer to non-compute-related measures. Examples include:

- BIOS-level case tamper-evident triggers (a compute-related measure).
- The siting of compute hardware in physical locations to which physical access is restricted.

- Padlocks on cases or racks in which compute hardware is sited.
- Biometric access controls to physical locations.
- Passive and active physical alarm systems.
- Security guards (see also clause 6.6).
- Smart card access controls to physical locations.

6.6 Personnel controls and checks

Other

Controls and checks (e.g. vetting/"clearing") on personnel are a standard security tool. These may include, for instance, credit, criminal record and background checks. The results of these may have an impact on the tasks that personnel are authorized to carry out, locations that they are authorized to visit - accompanied and unaccompanied - and information they are authorized to access.

Note that in some use cases and deployment scenarios, providing relevant clearance and authorization for all personnel who may come into everyday contact with a system, and extra checks and supervision for infrequent users, may provide adequate security to allow the number of other measures required to be reduced significantly. In other deployment scenarios, it is not feasible to clear all administrators with access to hosting systems or hosted applications. In these scenarios, providing restricted access based on the level of clearance is the only means to ensure security. Indeed, until artificial intelligence makes significant inroads in systems and network management, the ultimate application of the multi-layer administration principles described in the present document is enabling this personnel access segregation in large networks.

The areas within an NFV deployment which could be compromised by a person with malicious intent (or simply through incompetence) are many and varied, and availability of measures to detect such compromises also varies greatly. Some key points include:

- Hosting service administration.
- VIM administration.
- SDN administration.
- VNF catalogue administration.

Threat analysis for particular deployments will need to have considered the different attacker profiles (from gross insider/outsider considerations to granular role-based capabilities), the attack surfaces exposed to each, and appropriate defensive and forensic measures to be applied.

As in compute-based security measures, the principle of least privilege (and the closely related "need-to-know" principle in the case of personnel) is the foundation of a well thought-out security posture.

Note that available services available clearing and vetting staff and available services vary between countries. Best practice should be followed for the appropriate jurisdictions in which services are being operated.

6.7 Logical authentication controls

Hosting service

Logical authentication controls are software-based controls which may be applied to various processes and systems. They include, for instance, passwords, two-factor authentication (2FA), time-based controls. Where there is a physical component (e.g. a smart card), but the system making the decision on authentication and to which control is being authorized is software-based, then these are considered in this context as "logical", rather than "physical".

6.8 Read-only partitions

Hosting service

Read-only partitions are partitions which are never mounted with write permissions. This allows assurances to be made about the immutability of the data - which may contain executables - which resides on them. Although not a root of trust itself, if the contents of a read-only partition have been checked at an earlier point in time, they can form part of a chain of trust. The key benefits a read-only partition brings is that as long as it remains mounted read-only, it is immune to run-time changes which may impact on the operations and general levels of trust in the system.

Note that partitions may be mounted infrastructure layer (e.g. as a remote service), at the hosting service layer or at the virtualisation (hosted application) layer. The permissions at these layers may be different, and it is the accessing entity to whom a partition may be read-only, although other entities may have different permissions.

6.9 Write-only partitions

Hosting service

Write-only partitions are disk partitions which are, from the point of view of at least one actor, only writeable, and cannot be read. They are typically used for logging and audit purposes, where the authorizations of one actor are different from those of another. They may be implemented using local disks - where actors' capabilities will need to be carefully managed, as certain administrative rights typically allow re-mounting with changed permissions - or remotely, though availability may be an issue.

Note that partitions may be mounted infrastructure layer (e.g. as a remote service), at the hosting service layer or at the virtualisation (hosted application) layer. The permissions at these layers may be different, and it is the accessing entity to whom a partition may be read-only, although other entities may have different permissions.

6.10 Policies for workload placement

Other systems

A number of components in any NFV deployment are likely to have requirements for particular placement within the network. This may be for a variety of reasons, including:

- Logical or physical network proximity to physical systems.
- The need to be sited on the same host as one or more other components (often referred to as "affinity").
- The need to be sited on a different host to one or more other components, typically to avoid overuse of one or more resources such as bandwidth or CPU (often referred to as "anti-affinity").

Some hosted applications may also have requirements on their workload placement for security reasons: this is the case for all of the use cases identified in clause 4, whose requirements will be met with systems addressed in one or more of the approaches addressed in clause 7. As well at the requirements noted above, other requirements may include:

- placement on hosting services with particular capabilities;
- placement within particular geographic locations.

As important as the initial placement of workload component are policies regarding the migration or tear-down of such components and controls on copying of their state for debugging or testing.

The entity controlling placement (sometimes referred to as "scheduling") of the VMs, Containers or other components that comprise such hosted applications will vary depending on the architectural approach taken, and there may be one or more entities validating their placement, which may sit in different trust domains to the scheduling entity. This issue is considered generally outside the scope of the present document.

6.11 Communications Security

Hosting service, hosted application, other systems

Communications security is the set of techniques and measures associated with providing security for communications in and out of a particular component. As well as confidentiality and integrity, other measures may be important, including timeliness, bandwidth, delay and jitter. Note that traffic analysis is a complex and mature field, and a combination of different sets of data, combined with side channel-derived information, may compromise communications security even when measures are applied which might be considered sufficient in other contexts.

6.12 Measured boot

Hosting service

A measured boot is a process whereby a host completes up a boot sequence for which a chain of trust (CoT) is created by taking and recording measurements of the next component in the boot chain before passing control to it (see TCG PC Client Specific Implementation Specification for Conventional BIOS - Specification Version 1.21 Errata [i.11], clauses 1.3 and 3.1). A CoT needs to be anchored in a Trusted Building Block (TBB), which includes a TPM or an equivalent implementation that includes the protection capabilities, as provided by TPM (clause 6.17), and a Core Root of Trust for Measurement (CRTM) (see TCG PC Client Specific Implementation Specification for Conventional BIOS - Specification Version 1.21 Errata [i.11], clause 1.2.3). The CRTM should be the first component to be executed, which takes the first measurements and should only be updatable, if at all, through an authorized mechanism. The defining properties of the CRTM combined with the TPM's special features for measurements storage and reporting create the initial trust state upon which the CoT is built using the measured boot process.

The standard technique to take measurements is to calculate hashes of the various components and store these in the TPM's Platform Configuration Registers (PCRs). Sometime after the boot process is complete, a trust assertion can be performed based on the recorded measurements, a process called attestation.

Measured boot can be used standalone or complementary fashion with Secured boot ETSI GS NFV-SEC 003 [i.2], clause 4.4.5.1 with the note that a measured boot may complete even though some of the measurements do not produce the expected values, while the deviation, is, of course, logged.

6.13 Secured boot

Hosting service

A secured boot is a process where the integrity of various components in a boot sequence have been measured and found to be either:

- in accordance with expected values or;
- within tolerable ranges for the required host profile.

In contrast to measured boot, if the criteria for secured boot are not met, the boot does not complete, and the deviation, is, of course, logged.

See Network Functions Virtualisation (NFV); NFV Security; Security and Trust Guidance:

- ETSI GS NFV-SEC 003 [i.2], clause 4.4.5.1 for further details.

6.14 Concealed resource usage

Hosted application

Constant resource usage is a measure whereby the host application makes use of certain resources at all times in order to conceal from other parties the operations that it is carrying out - or not carrying out - at a particular time. It is considered the most effective approach to mask operations, but can be costly in terms of resource usage. This cost is likely to figure particularly in cases where the hosted application is not itself providing revenue-positive services for the operator of the hosting service. What is more, constant usage of one particular resource may well not be enough to mask operations, as monitoring of other channels may also reveal information to an attacker. For instance, even if a hosted application is employing constant amounts of CPU and network resource, changes in storage usage over time may well leak information.

An alternative to constant resource usage is to conceal usage through masking true usage over time by injecting statistically similar activity, and taking steps to divorce usage of different types, where possible, over time, so that extrapolation of information garnered across channels to infer operational usage, though not rendered impossible, becomes significantly more difficult, and requires access to various data and metadata which are unlikely to be available to a single attacker. In this case, truly constant usage is not correct.

One alternative to constant usage when real-time information is not required is "bursting" of resource usage. This may be of particular utility where networking resources are scarce, and so true constant resource usage is impossible. However, the sudden appearance of traffic on the network (to satisfy a Lawful Interception request, for instance - see clause 4.5) may be unacceptable It may, however, be acceptable to have short bursts of information, particularly when network usage is otherwise low - for instance during off-peak hours. If such bursts take place whether or not true activity is taking place (with carefully prepared mimicked data, for instance), then the *average* usage of resources will remain the same. The same applies to usage of CPU resources for processing of data streams where the amount of storage available is sufficient. It should be noted, however, that coordinated analysis of various side channels (storage I/O, RAM usage, CPU usage, network usage) may, on occasion, still leak sufficient information to be useful to an attacker, so such approaches need to be considered with great care.

Another alternative is to make informed decisions about the amount of information that can be garnered from various channels, and to introduce cross-channel independent fluctuations sufficient to mask true operation, though, as noted above, to do this reliably and efficiently is extremely complex.

6.15 Attestation

Hosting service, hosted application, other systems

Detailed information around attestation is provided in ETSI GR NFV-SEC 007 [i.8], any approach to attestation should consider the issues raised in clauses 5.0.3, 5.0.4, 5.1.1, 5.1.3, 5.1.4, 5.1.6, 5.2.10 and 5.3.3. Typically, a TPM (clause 6.17) is a core measure associated with attestation on the hosting service.

6.16 Hardware-mediated execution enclaves

Hosting service, hosted application

A hardware-mediated execution enclave is defined as an area of process space and memory within a system environment within a computer host which delivers confidentiality and integrity of instructions and data associated with that enclave. This enclave is protected from eavesdropping, replay and alteration attacks as the programs within the enclave are executed.

An enclave is considered capable of executing processes, and executable code can be loaded into it. Various capabilities may be provided by such an enclave, but at minimum, the following are expected:

- The ability for executable code to be loaded into the enclave.
- The ability for the host to attest to the integrity of the executable code prior to execution.
- The ability to load data into the enclave.

- The ability to execute code within the enclave without other processes on the host being able to inspect, alter or replay the instructions or associated data. Note that these protections are not just against unprivileged processes, but also against the HBRT (e.g. TPM, HSM, etc.) and hypervisor processes which may be running at an escalated privilege level.

6.17 Trusted Platform Module (TPM)

6.17.0 Introduction

Hosting service, other systems

Note that there is some overlap in functionality between TPMs and HSMs: see also clause 6.20.

A Trusted Platform Module (TPM) is a hardware cryptographic module that can securely store sensitive data and perform various cryptographic operations. Authentication (a process to prove the identity attribute of an entity, i.e. the TPM acting as the integrity reporting entity) and attestation (a process that enables the software integrity state to be reported and verified in order to determine its trustworthiness) are necessary steps to ensure trusted computing. A TPM can authenticate itself using the credentials stored in the shielded memory and provide integrity measurements reports to prove platform software is trustworthy.

The nature of a TPM's shielded memory ensures that information may be stored and protected from external software attacks. A variety of applications storing data and secrets protected by a TPM can be developed. These applications make it much harder to access information on a computing platform without proper authorization. If the software configuration of a platform has changed as a result of unauthorized activities, access to such data and secrets can be denied.

Various TPM specifications exist: the most recent at time of writing is TCG "TPM 2.0 Library" [i.14].

TPMs can provide a hardware root of trust on a hosting service platform, and can be leveraged for operations such as measured boot and attestation. The functionalities of TPMs can be achieved by equivalent hardware implementations that include the protection capabilities, as provided by TPMs.

Various other uses of TPMs in virtualised environments have been proposed. Two specific examples are worthy of consideration here, and can be characterized as "shared TPM" and "virtual TPM" (sometimes "vTPM"), though various terms are used to describe them in the industry.

For both of the examples, the lower layer (e.g. Hosting OS, etc.) needs to be explicitly trusted by the hosted application, as the former still maintains control over the capabilities provided: important security capabilities of the hosted application remain in the same trust domain as the hosting service.

6.17.1 Shared TPM

In a shared TPM use case, the TPM which is part of the hosting service's hardware platform is shared by one or more Virtual Machines (VMs) or Containers. This continues to provide a hardware-based trust model, but immediately raises the issue of trust domains.

Further sharing of the TPM with other VMs is fraught with complications, as there is no simple way to establish a single control entity.

A TPM Access Broker is typically used to control and synchronize multi-process access to a single Shared TPM.

Such control and synchronization are to ensure that when one process is in the middle of communication with the TPM (e.g. sending a command and receiving a TPM response), no other processes are allowed to communicate with the TPM (e.g. sending a different TPM command). Another function of the TPM Access Broker is to only grant access to TPM sessions, objects, and sequences to the process that owns them, thus providing confidentiality and access control.

TCG TPM Access Broker and Resource Manager are defined in this specification: TCG "TSS TAB and Resource Manager Specification" [i.15].

6.17.2 Virtual TPM

A "virtual TPM" (vTPM) is a software application provided either by the hosting service or directly by a Virtual Machine Manager (VMM), anchored in a physical TPM (pTPM). To a hosted application, a vTPM provides similar functionality as a pTPM does for the hosting service.

There are two models in which the vTPM interacts with the pTPM:

- The vTPM implements most of the TPM API used by the hosted application, but it also allows a limited number of commands to be passed directly to the pTPM. This design is useful for retrieving the hosting service software integrity state from within the hosted application, possibly for creating an aggregated integrity report (hosted application state and hosting service state). The vTPM application will further use the pTPM as a hardware-based Root of Trust (RoT) anchor, for example to protect its sensitive data while at rest.
- The vTPM implements all the TPM API used by the hosted application and the pTPM is never used directly by the hosted application. In this design the pTPM is only used for anchoring the vTPM application to a hardware-based RoT.

The following properties of a trusted platform are implicitly constrained in a physical TPM with a hardware RoT and are not implicitly constrained in a virtual TPM:

- Roots of Trust (e.g. Root-of-Trust for Measurement (RTM)) are implemented in a virtual TPM differently from a physical TPM.
- A virtual TPM typically has more resources than a physical TPM (e.g. more Platform Configuration Registers (PCR)).

The physical protection available to a virtual TPM differs depending on its hosting service. Trust in the virtual TPM depends on its hosting service. Hence there is a need for the host to attest its trustworthiness (e.g. by providing measurements, evidence, and provenance of host certification) to the Virtual TPM Manager prior to the instantiation of a virtual TPM; "TCG Virtualised Trusted Platform Architecture Specification" [i.17].

6.18 Self-encrypting drives/storage

Hosting service, other systems

Self-Encrypting Drives (SEDs) provide hardware-based data security and the ability to render data unreadable very quickly (e.g. for removal of a user account or device retirement or refurbishment). Self-encrypting drives encrypt data using a key as it is written to the disk, then decrypt it on read. Self-encrypting drives support multiple Logical Block Arrays (LBAs) each with a separate key (e.g. for separate access control of OS boot images and user installed applications). Deletion of an SED encryption key renders the data on the entire drive or a specific LBA range unreadable, completely eliminating the need for data-overwrite and simplifying storage management.

SEDs provide the following capabilities:

- data confidentiality even if the drive is physically removed by an attacker;
- operating system and user application data is always encrypted at write time;
- once the SED is provisioned, encryption is automatic and cannot be deselected;
- erasure of SED keys makes encrypted data immediately unreadable;
- CPU resource conservation since the encryption workload is handled by the SED itself.

6.19 Direct Memory Access to hardware resources

Hosting service, hosted application

Direct Memory Access (DMA) is a technique whereby certain hardware resources can be assigned directly to Virtual Machines (or, specifically, to areas of RAM associated with particular processes). It is typically used to provide performance improvements, but can also provide security isolation capabilities, as the hosting service is not mediating access to these hardware resources, so, in the case of a Network Interface Card (NIC), for example, incoming and outgoing packets are not managed by the hosting service hypervisor or Operation System. Although this means that there may be somewhat higher levels of confidentiality, integrity and availability than in the non-DMA case, since the hosting service can:

a) inspect the memory of the hosted application (see clause 6.1); and

b) manipulate CPU cycles assigned to the hosted application.

These improvements are only relative.

6.20 Hardware Security Modules

6.20.1 Introduction

HW Security Modules (HSM) provide physical and logical protection for data and in particular for cryptographic material, such as keys. In addition they offer highly secured cryptographic services, with physical and logical protection.

6.20.2 Physical Hardware Security Modules

Physical HSMs are fully contained solutions for scalable cryptographic processing, key generation, and centralized key storage. As purpose-built appliances, they automatically include the hardware and firmware (i.e. software) necessary for these functions in an integrated package. They may be optimized for a specific or general purpose (e.g. performance, environment, portability, interface).

HSMs cooperate with services/applications via several secure means (interfaces, protocols).

Functions supported by HSMs include:

- Life-cycle management of cryptographic keys.
- Cryptographic processing which produces the dual benefits of isolating and offloading cryptographic processing from application servers.

Physical and logical protection of the appliance is supported by a tamper resistant/evident shell; and protection from logical threats, depending on the vendor’s products, is supported by integrated firewall and intrusion prevention defences. Some HSM vendors also include integrated support for two-factor authentication.

Security certification (e.g. PCI-HSM, NIST FIPS 140-2 [i.16], Common Criteria) is typically pursued by HSM vendors and positioned as a product feature.

In context of multi-layered host administration, an HSM may be attached to an NFVI as a HW anchored security root, e.g. to provide cryptographic services or to support implementation of secure storage. HSMs require specific access protection (see clause 5.2.7 Secure storage, PKI, HSS). HSMs allow the storage of keys in a single unit and centralize the key management which constitutes a consistent layer of key protection and decreases the risk of keys being compromised.

Note that HSMs are usually distinguished from HW TPMs (which are optimized for attestation services at platform level, but may offer specific crypto services as well - see clause 6.17 Trusted Platform Module (TPM)) by their characteristic of high-scalability in key management.

6.20.3 Virtual Hardware Security Modules

In multi-tenancy use-cases, virtual HSM (vHSM) are used to provide the HSM functionality for each tenant, using a single or several physical HSM. Each vHSM is isolated between each other with distinct access control, cryptographic domains, policies, memory and time separation, etc. The isolation is done on a per-vHSM basis and may be a logical separation or a physical separation where dedicate hardware resource are allocated per vHSM.

6.21 Software integrity protection and verification

SW integrity protection and verification encompasses a variety of measures to detect unwanted modifications of (static) software, data or firmware compared to a reference model. As such it has applicability at various points in the lifecycle of a system, including provisioning, boot and run-time. Usually it is based on cryptographic hash values calculated for a defined SW module, file, image (VM), etc., or memory content. The hash values need to be trusted, which can be achieved by several means.

One widely applied method is to protect software with digital signatures using a private key (of an authoritative source, e.g. SW vendor or community) for signature creation and an associated public key (e.g. RSA, and/or involving PKI, certificates) for verification. Signed software (or data) allows proof of origin as well as integrity and its protection are independent from storage or delivery media, and does not need specific hardware. However, public keys/certificates need to be protected against exchange in the verification system. Verification of signed software (in a trusted system) allows local decisions on integrity status of a protected object and does not require extra infrastructure and external communication.

Note that this measure is applicable to both the hosting service and the hosted application, and different trust relationships and implementations may be required for these different logical components. It is also very relevant to the provision of those parts of a hosted application which may execute or act as data within a hardware-mediate execution enclave (see clause 6.16).

This technique requires a trust relationship to be established with one or more other components which might typically sit in the same trust domain as a remote attestation authority (see clause 6.15), though any component providing software integrity protection and verification services should consider the issues raised in clauses 5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 5.1.6, 5.2.10. It should also be noted that not all information should necessarily be made available to all parties: the existence or state of some software components may be sensitive in and of itself.

Depending on the capabilities of a particular attestation authority (see clause 5.3.3), it may provide all or some of the measures associated with this issue.

7 Technical approaches to multi-layer administration

7.0 Introduction

These approaches to providing architectures and solutions to meet the requirements of the use cases noted above are *informative.* They will contain a mix of logical, physical, process and other measures to address the various requirements. Several points are important to note:

- No system should be considered completely secure against an adequately resourced and motivated attacker.
- All approaches to security are an attempt not to remove risk, or even necessarily to reduce it, but to provide systems where the risk can be quantified, qualified, mitigated and managed.
- The approaches described above are examples only, and building a solution which meets them does not guarantee that the solution will be fit for purpose in any particular situation or deployment.

In all cases, establishing and maintaining an appropriate chain of trust is a *sine qua non* without which the true level of trust in the security of the system cannot be ascertained. The establishment is not solely at boot time of the hosting system, but also at its provision: the same goes for the hosted application. In order to be able to maintain levels of trust, however, run-time checks are also required on all aspects of the system. This includes not only the software, but also, given the increasing use of hot-pluggable systems, certain aspects of hardware as well.

7.1 Approaches to address specific requirements

This clause describes various approaches to providing secure architectures for various use cases. It does not attempt to map these approaches to the specific use cases, as an exact mapping between the various requirements identified in clause 4 and the measures identified in clause 6 depends on implementation, and the present document takes in informative, rather than normative, approach. The larger the set of requirements identified, however, the more advanced and complex are likely to be the measures needed to meet them. The aim of this clause is to show that, even in the absence of these more complex measures, architectural approaches may be available to service at least some of the use cases noted.

A key point, however, is that the negative operational impact of implementing the architectural approaches employing the simpler measures will be significant: such approaches trade off simpler micro-system architectures with much more complex macro-system architectures.

7.2 Generic approaches

7.2.0 Basic comparison

See clause 8 for a description of the key differences between the three approaches.

Table 11 notes the various measures which may be used in the different approaches. Those measures which are considered best practice are not necessarily noted as core to an approach, as they may be considered unnecessary in some deployments.

Table 11

	7.2.1 Single, restricted hosts	7.2.2 Pooled, restricted hosts	7.2.3 Pooled, unrestricted hosts
6.1 Memory inspection		M	M
6.2 Secure logging	M	M	M
6.3 OS-level access control		X	M
6.4 Post-incident analysis	M	M	M
6.5 Physical controls and alarms	X	X	M
6.6 Personnel controls and checks	X	X	X
6.7 Logical authentication controls	X	X	X
6.8 Read-only partitions		M	M
6.9 Write-only partitions		M	M
6.10 Policies for workload placement	X	X	X
6.11 Communications Security	X	X	X
6.12 Measured boot	M	X	X
6.13 Secured boot	M	X	X
6.14 Concealed resource usage		M	X
6.15 Attestation	M	X	X
6.16 Hardware-mediated execution enclaves			X
6.17 Trusted Platform Module (TPM)	M	X	X
6.18 Self-encrypting drives/storage		M	M
6.19 Direct Memory Access to hardware resources		M	M
6.20 Hardware Security Modules		M	M
6.21 Software integrity protection and verification	X	X	X
NOTE: X - core to approach. M - may be needed.			

7.2.1 Single, restricted hosts

Physical controls, restricted personnel, trusted single hosts

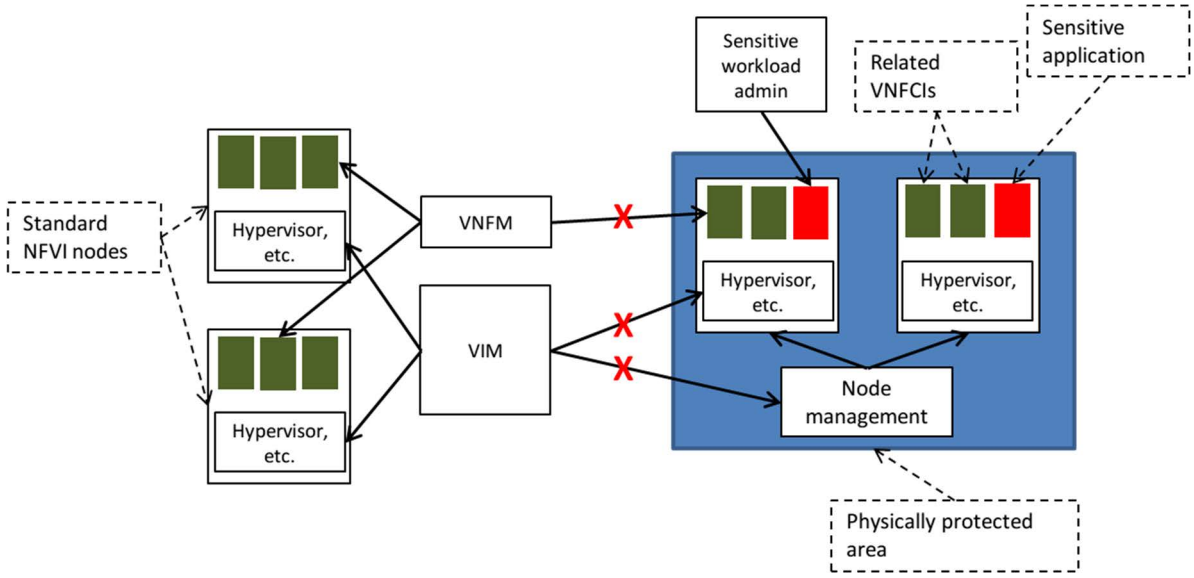

Figure 1: Single, restricted hosts: example deployment diagram

The least complex approach to managing restrictive requirements such as those identified in the previous clauses is to restrict the number of hosting services - NFVI hosts, in ETSI NFV nomenclature - which offer certain capabilities, and to meet other requirements not through technical measures but by physical and procedural controls. The simplest approach is provisioning a small number of dedicated hosts on which all hosted applications requiring these specific capabilities run. All traffic for services requiring these services will therefore need to pass through these hosts, so, for the example of Lawful Interception, all voice and data traffic would need to pass through these NFVI hosts. Although this approach may seem architecturally simple, siting a few hosts in one location - at the edge of the network - is not sufficient. Some services - for example VoLTE - cannot be sampled in this way as the nature of the traffic passed is such that it cannot be comprehensively re-composed at this point. There is a trend taking place within operators - separately to NFV, but accelerated by it - to distribute Network Functions at various locations within the network, and choosing the approach suggested here means either that this trend will need to be reversed or that multiple sites of restricted hosts will need to be made available.

One key control is to restrict the administrators of these NFVI hosts to authorized personnel, and to restrict physical access to these servers. However, these controls are not enough if these hosts are to exist within the NFV deployment, as a number of the functions of the Management and Orchestration domain may also touch the NFVI host, including:

- networking resource metadata;
- RAM resource metadata;
- CPU resource metadata;
- vSwitch control plane;
- vSwitch telemetry information;
- storage and storage metadata.

Exposure of any one of these may break the requirements of the use case being addressed, and it is therefore likely that such an approach requires that the hosts are logically removed from the scope of the NFV deployment: they are not truly NFVI hosts at all, and many of the benefits of NFV are therefore lost.

A key point to note is that a vulnerability at the hosting hypervisor or OS layer may lead to a *greater* exposure in this approach than in an approach where each host is more trusted, because the protection level has been raised not by technical measures (by increasing the trust that can be placed in the hosting service) but by procedural and process measures. The attack surface, though reduced by physical measures, is increased in logical terms, and making significant inroads in reducing it, given the requirements for control, orchestration and management of the non-sensitive applications on the hosting service, is likely to be extremely onerous.

The major disadvantage of this approach is that it will need to be implemented on ***all*** gateways and as the number of gateways increases, this becomes less and less feasible. Added to the point that this is not truly integrated with an NFV deployment, this approach is ***not*** considered to be operationally workable.

Benefits and disadvantages

Benefits:

- Fewer technical measures required:
  - there may be some reduction in the number of technical measures required over a fully virtualised implementation (though see "Disadvantages" below).
- Existing models of control require fewer changes than other approaches:
  - the models of control used in existing deployments are well understood and can be extended to some extent.

Disadvantages:

- Administration split:
  - NFVI domain administration needs to be restricted to authorized users.
  - Management and Orchestration administration needs to be restricted to authorized users and segregated from other components, requiring duplication of systems and lack of single database of record.
- Inapplicable to all services:
  - a growing number of services (e.g. VoLTE) do not present intelligible data at the network edge, and so would require a separate set of hosts to be provided: there is no single point in the network where all data will be sampleable.
- Special measures for hypervisor & OS-layer required:
  - Although administrative users may be authorized, if standard virtualisation techniques are used, the hypervisor (or Container) host and all OS-layer software will need to be carefully audited and controlled. A vulnerability within these layers, particularly if it allows host privilege to be attained by an application-level entity, presents not fewer risks but *greater* risks than in the usual case, as more sensitive data is available in an unprotected domain. Remediation may also be significantly more difficult, due to the lack of centralized control.
  - There is little reduction in the attestation requirements on these hosts, even though the capabilities they are required to provide may be reduced. The overhead of auditing and managing a small number of servers may not be significantly lower than that for a larger number.
- Non-core traffic:
  - If all service traffic will need to pass through a small number of hosts, opportunities for providing non-core routing services via products such as vCPE and Mobile Edge are reduced.

Benefits of NFV not available:

- Quasi-physical architecture:
  - Network will need to be architected as if composed of physical implementations of each network function that is affected.

- Host software upgrade:
  - Options are fewer, as the number of available hosts to provide services is bounded to the set of restricted hosts.
- Host hardware maintenance:
  - Options are fewer, as the number of available hosts to provide services is bounded to the set of restricted hosts.
- Scale out:
  - Opportunities for expansion and shrinking of services is bounded by the number of hosts in the restricted set.
- Migration:
  - The ability to migrate hosted applications or portions of hosted applications is severely curtailed.
- Bandwidth availability:
  - If a small number of hosts is chosen, opportunities for parallelising of traffic are reduced, and economies of scale more difficult to leverage.
- Restriction of SDN options:
  - If all service traffic has to pass through a small number of hosts, the benefits of SDN are reduced.
- Significance of routing controls:
  - If all service traffic has to pass through a small number of hosts, routing controls *outside* the restricted hosts and their servicing network fabric will also need to be carefully controlled.
- Single point of failure:
  - Agility of service provision is restricted in event of serious attacks or disasters, as a physical point of failure remains in the network.

7.2.2 Pooled, restricted hosts

7.2.2.0 General case

Physical controls, restricted personnel, trusted pools

The case where there is a pool - or a set of pools - of restricted hosts brings some advantages over the case described above (in clause 7.2.1), but does not remove all of the disadvantages. This clause examines the case where there is only one pool, though the case where there are more may be similar: trust relationships for multiple pools may be complex. A key point to note is that there is no assumption that all hosts in a pool are necessarily sited in the same physical location. A "pool" is a logical agglomeration, not a physical one, and it is possible to consider multiple hosts, all physically secured, but all in separate locations on the same or separate sites, to be members of a single pool. In order to ensure that a particular host should be considered a member of a pool, attestation is expected to be an important measure to consider within this case (see clause 6.15).

The key differentiator between this case and the unrestricted case (described in clause 7.2.3) is that physical controls are still required as the hosts in the pool are not able to meet all the requirements on them without further measures: specifically, where memory inspection by the hosting service may impact on the hosted application. There are two expected "flavours" of pool:

1) pools where usage of resource concealment (6.14) is unavailable;
2) pools where usage of resource concealment is available.

Note that the capability to conceal resource usage is not necessarily solely a function of the hosting service, but may also be provided in whole or in part by the hosted application. In this case, what was initially considered a pool of type 1 may actually function as a pool of type 2, yielding different behaviour and advantages. See clauses 7.2.2.1 and 7.2.2.2 for further analysis.

In a pooled, restricted host environment, as in the case of single, restricted hosts, all traffic for services requiring these services will need to pass through these hosts. However, as a pool of hosts is available, and these may be in different physical and logical locations, this reduces the restraints on network architecture - again, both physical and logical.

The traffic inside the pool (but across the "open") network, and across pools in the case of multiple pools, creates a new attack surface, not existent in the single, restricted host case. Of course, this traffic is assumed to be correctly encrypted, but traffic analysis attacks are nevertheless possible.

As in the case of single, restricted hosts, a vulnerability at the hosting hypervisor or OS layer may lead to a *greater* exposure in this approach than in an approach where each host is more trusted, because the protection level has been raised not by technical measures (by increasing the trust that can be placed in the hosting service) but by procedural and process measures. The attack surface, though reduced by physical measures, is increased in logical terms, and making significant inroads in reducing it, given the requirements for control, orchestration and management of the non-sensitive applications on the hosting service, is likely to be extremely onerous.

7.2.2.1 Type 1 - no resource concealment

Where there is no resource concealment available, the extent to which the hosts in a pool of this type can be managed and orchestrated within the scope of an NFV deployment may be significantly restricted. This is because the functions used by a VIM to manage and orchestrate a host are likely to give information to unauthorized users in contrast to the requirements described in clause 5.2.5. Telemetry data typically gathered by the NFV VIM to provide information to VNFM, Orchestrator or EMS entities could provide inappropriate levels of information about the hosted applications. Note that although the amounts of data of any particular type may be considered small, aggregated data across multiple channels and over time may yield enough information to expose information beyond the acceptable levels.

A separate set management and operations components is likely to be needed for **each** sensitive function requiring such security measures, e.g. LI, HSS, Billing, Edge protection, etc. as they all operate in different trust domains. This increases the disadvantage that it will need to be implemented on all gateways and as the number of gateways increased, this becomes less and less feasible, as a variety of "shadow" deployments need to be deployed and operated. Added to the point that this is not truly integrated with an NFV deployment, this approach is **not** considered to be operationally workable, especially when scaling to real-world deployments.

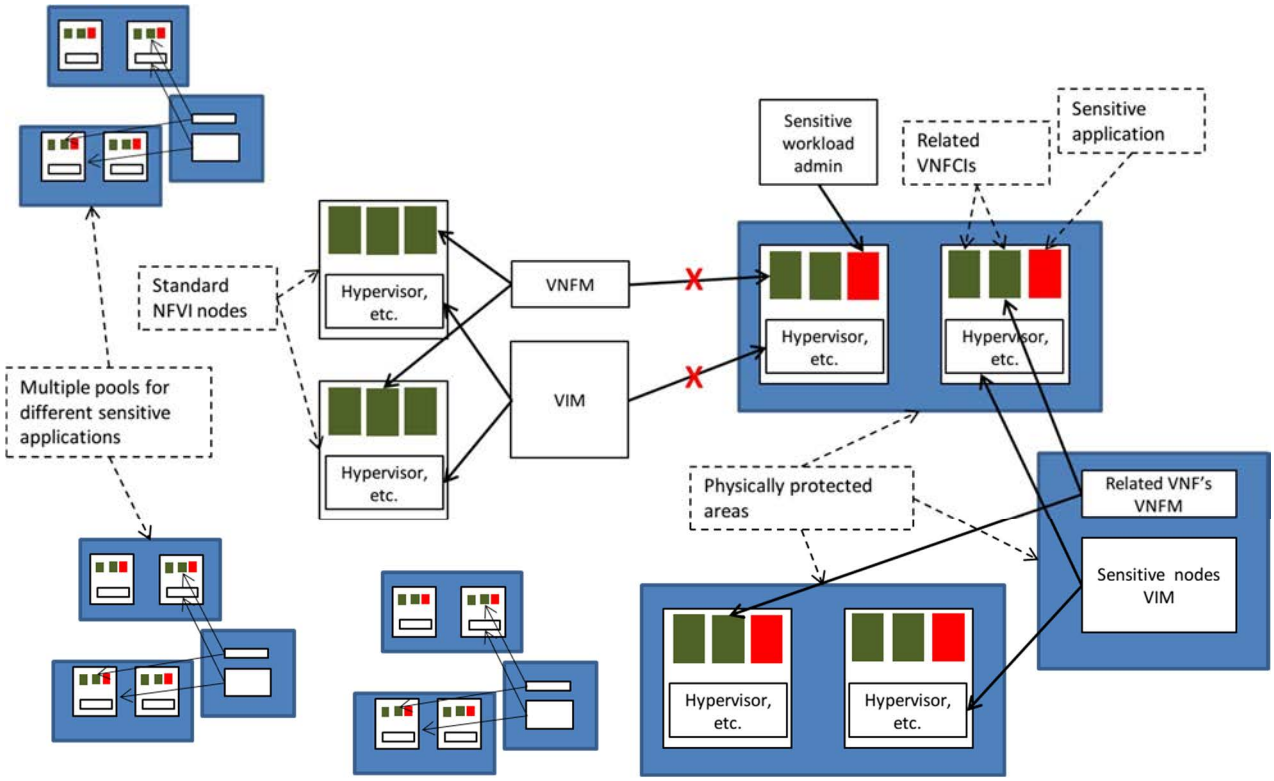

Figure 2: Pooled, restricted hosts, no resource concealment: example deployment diagram

7.2.2.2 Type 2 - resource concealment

Where resource concealment is available, it may be possible to allow the VIM to have some level of management and orchestration control over individual hosts, as it should be possible to restrict the data available to unauthorized users to that which cannot be used to infer sensitive information about hosted applications. Some capabilities of the VIM will need to be restricted: clause 5.3.2 notes some areas, but more generally, it may be impossible to migrate some hosted applications (or their components) as these may need to reside within the trusted pool(s).

Given that hosted applications within these restricted pools may, in the type 2 case, be under the control of a standard VIM - albeit with some restrictions - they can be considered part of the more general NFV deployment, but multiple management and operations components will still be required. This increases the disadvantage that it will need to be implemented on all gateways and as the number of gateways increased, this becomes less and less feasible, as a variety of "shadow" deployments need to be deployed and operated. Added to the point that there is still not complete integration with an NFV deployment, this approach is not considered to be operationally workable, especially when scaling to real-world deployments.

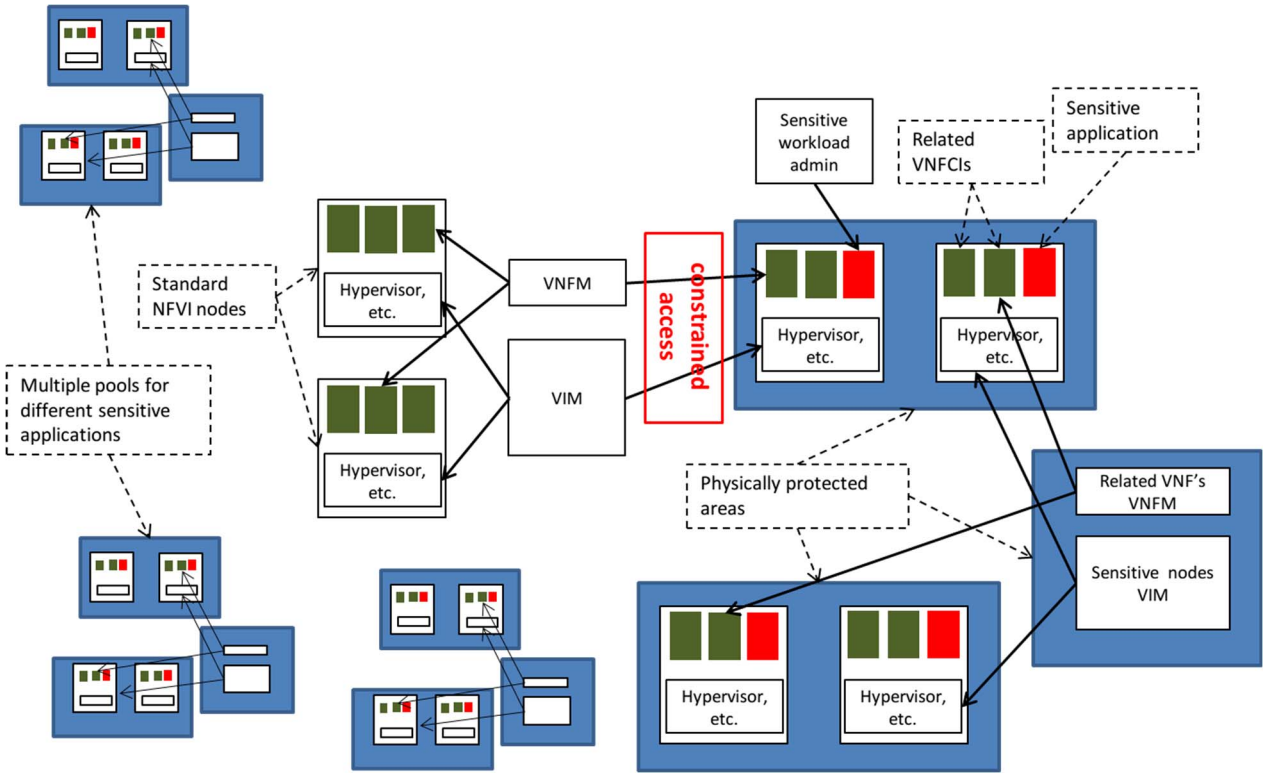

Figure 3: Pooled, restricted hosts, with resource concealment: example deployment diagram

Benefits and disadvantages

Benefits:

- Fewer technical measures required:
  - There may be some reduction in the number of technical measures required over a fully virtualised implementation (though see "Disadvantages" below).
- Logical architecture:
  - It is possible to break away from some of the physical restraints imposed by network architectures, as pools may be sited in appropriate physical and logical locations.
- Host software upgrade:
  - Host software upgrade is possible as hosted applications may be able to be transferred to different hosts in the same pool.
- Host hardware maintenance:
  - Options are increased, as hosted applications may be able to be transferred to different hosts in the same pool.
- Scale out:
  - Options are increased, as components of hosted applications may be able to be instantiated on different hosts in the same pool.
- Migration:
  - The ability to migrate hosted applications or portions of hosted applications is increased to include hosts in the same pool.

Disadvantages:

- Administration split:
  - Depending on the level of resource concealment available, the following disadvantages may have higher or lower weight:
    - Some or all NFVI domain administration may need to be restricted to authorized users.
    - Some or all Management and Orchestration administration may need to be restricted to authorized users and segregated from other components, requiring duplication of systems and lack of single database of record.
- Special measures for hypervisor & OS-layer required:
  - Although administrative users may be authorized, if standard virtualisation techniques are used, the hypervisor (or Container) host and all OS-layer software will need to be carefully audited and controlled. A vulnerability within these layers, particularly if it allows host privilege to be attained by an application-level entity, presents not fewer risks but *greater* risks than in the usual case, as more sensitive data is available in an unprotected domain. Remediation may also be significantly more difficult, due to the lack of centralized control.
  - There is little reduction in the attestation requirements on these hosts, even though the capabilities they are required to provide may be reduced. The overhead of auditing and managing a small number of servers may not be significantly lower than that for a larger number.

Benefits of NFV not available:

- Quasi-physical architecture:
  - Logical network architecture is still somewhat constrained.
- Host software upgrade:
  - Options are fewer, as the number of available hosts to provide services is bounded to the set of restricted hosts.
- Host hardware maintenance:
  - Options are fewer, as the number of available hosts to provide services is bounded to the set of restricted hosts.
- Scale out:
  - Opportunities for expansion and shrinking of services are bounded by the number of hosts in the restricted set.
- Migration:
  - The ability to migrate hosted applications or portions of hosted applications is somewhat curtailed.
- Bandwidth availability:
  - If the pool size is small, opportunities for parallelising of traffic are somewhat reduced, and economies of scale more difficult to leverage.
- Restriction of SDN options:
  - If all service traffic has to pass through the pool, the benefits of SDN are somewhat reduced.
- Significance of routing controls:
  - If all service traffic has pass through the pool, routing controls *outside* the restricted hosts and their servicing network fabric will also need to be carefully controlled.

7.2.3 Pooled, unrestricted hosts

Fewer physical controls, restricted personnel, trusted pools, placement and movement policy

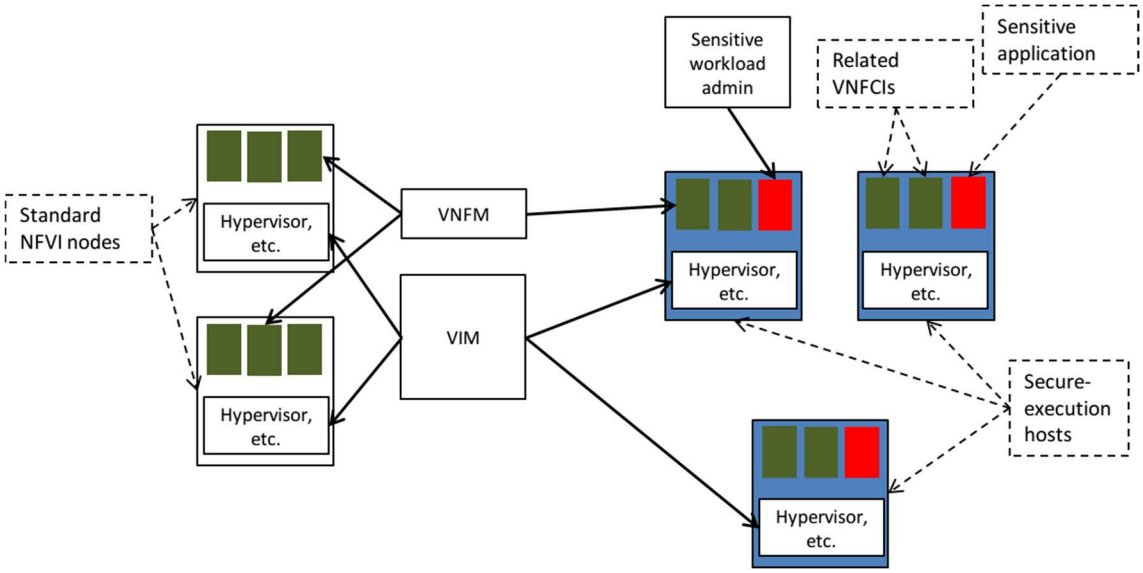

Figure 4: Pooled, unrestricted hosts: example deployment

The key differentiator between this architectural approach and those preceding it is the availability of trusted hosts which do not require physical access controls beyond the standard security for hosts providing sensitive services: specifically, hosts where host administrators are not restricted in their logical access to the host platform. For the purposes of the present document, these hosts are referred to "secure-execution hosts", though it is noted that, as in all contexts, the security of these hosts is relative: there is no completely secure host expected to be described. It is likely that only a subset of available hosts within any specific NFV deployment will be secure-execution hosts and therefore appropriate for the placement of sensitive hosted applications, but rules would be enforced as to placement of these workloads, and these rules would be auditable and subject to careful change control. The detail of this process is out of scope for the present document.

This imposes the fewest constraints on the physical location of secure-execution hosts other than the geographical and physical requirements imposed by particular hosted applications, which are best managed by location attestation. Addition of further secure-execution hosts is therefore significantly simplified over the cases described in clause 7.2.2, and as long as hosts have appropriate hardware capabilities, they can be added and removed from the pool of hosts as required, given authority to change by appropriate entities.

Core properties of secure-execution hosts are:

- resource concealment available: allows integration with Management and Orchestration trust domain;
- full multi-layer administration: hardware-based ability to provide confidentiality of:
  - data, data-related metadata, process data, process-related metadata;
  - secure communications;
- attestation (may include location attestation);
- secure clean-up;
- availability of entropy.

Secure-execution hosts provide the ability for hosted applications to run within a hardware-mediated secure enclave, requiring specific hardware capabilities, and for authorized administrative entities to communicate securely with them without the risk of unauthorized access. Secured routing and switching may be required as part of the provision of secure communications. Resource concealment allows hosted applications which require it to conceal the amount of resources that they are consuming from unauthorized entities.

Since resource concealment is available, telemetry can be shared with Management and Operations components, and since multi-layer administration is available, the host can participate fully within the NFVI (or as part of a pool where only a subset of hosts is secure-execution hosts). Administrative functions - whether human-managed or automatic via the VIM - may be applied as normal for an NFVI host.

The measures required for a secure-execution host include the following:

- personnel controls and checks: only specific staff to be allowed access to controlled resources;
- logical authentication controls: only specific entities to be allowed access to controlled resources;
- policies for workload placement: workloads (VMs or Containers) requiring a secure-execution host only to be instantiated on or migrated to such a host;
- communications security: hosted applications executing within the secure enclave of a secure-execution host need to be able to communicate securely with other entities on and off the same host;
- measured and secured boot: the secure-execution host to have the ability to assert its integrity and capabilities of the relevant entities;
- Trusted Platform Module: a hardware TPM or an equivalent hardware implementation that includes the protection capabilities as provided by TPM to be available as a hardware root of trust;
- attestation: the secure-execution to be attested by a relevant entity;
- concealed resource usage: hosted applications which require the ability to conceal their resource usage from unauthorized entities to be able to do so;
- hardware-mediated execution enclaves: execution of hosted applications within a hardware-mediated execution enclave.

Benefits and disadvantages

Benefits:

- All benefits of NFV are available as all hosts become part of secure-execution pool(s).
- Any hardware-capable host can be added to the full NFVI deployment.

Disadvantages:

- Specific hardware may be required.

Benefits of NFV not available (all minimized as pool deployments are expanded and network scaling advantages are realized):

- Quasi-physical architecture:
  - Logical network architecture may be somewhat constrained.
- Host software upgrade:
  - Options are fewer, as the number of available hosts to provide services is bounded to the set of restricted hosts.
- Host hardware maintenance:
  - Options are fewer, as the number of available hosts to provide services is bounded to the set of restricted hosts.
- Scale out:
  - Opportunities for expansion and shrinking of services are bounded by the number of hosts in the restricted set and their relationship to the other NFVI hosts.

- Migration:
  - The ability to migrate hosted applications or portions of hosted applications is somewhat curtailed.
- Bandwidth availability:
  - If the pool size is small, opportunities for parallelising of traffic are somewhat reduced, and economies of scale more difficult to leverage.
- Restriction of SDN options:
  - If all service traffic has to pass through the pool, the benefits of SDN are somewhat reduced.
- Significance of routing controls:
  - If all service traffic has to pass through the pool, routing controls *outside* the restricted hosts and their servicing network fabric will also need to be carefully controlled.

8 Roadmap to secure-execution hosts

8.0 Applicability of secure-execution hosts

The first point to highlight in this clause is that not all of the virtualised components within an NFV deployment are VNFs, in the sense that they are parts of the service chain. Other components such as those within the Management and Orchestration domain may also benefit from being placed on a secure-execution host. A non-exhaustive table of some components which are candidates for placement on secure-execution hosts is provided below.

Table 12

Node Class	Network Technology	Element/Function Names
Lawful Interception (POIs and supporting nodes)	3GPP Fixed Networks	P-CSCF, S-CSCF, IBCF, MSC, GMSC, HSS, HLR, VLR, SGSN, GGSN, S-GW, PDN-Gateway, IMS-ALG, MME, MFRP, MRFC, SBC, BMSC BRAS, DSLAM, Local Exchange, Trunk Exchange, AAA Radius functions, SBC, Border Gateways.
Subscriber data/Billing Databases	3GPP All	HSS, VLR, HSS, SMSC OSS/customer billing systems, PCI (Payment card industry designated system).
Security perimeter boxes	All	Firewalls, Switches, Gateways (CS and Internet), Proxy servers, Monitoring/active filtering equipment.
Cryptographic Functions/CAs	3GPP All	AUC Network Root CAs.

This clause does not attempt to provide normative guidance on how to build an architecture to address any of the approaches identified in clause 7, rather, it addresses some of key differences between them, and how some of the measures noted in clause 7.2.0 might be used.

8.1 Moving to single, restricted hosts

The key measures required for single, restricted hosts are physical, personnel and logical controls: all techniques which are standard for sensitive telecommunications components today. The difference, however, is that where, for non-virtualised deployments, these measures are applied to the components themselves, in a virtualised (NFV) deployment, they will need to be applied not only at the component level, but also at the hosting service ("host", "system" or "NFVI node") level, since unauthorized access at this level can also lead to compromise at the component level.

These measures would typically be performed by placing those hosting services on which sensitive components will run within secured, locked rooms or cages (physical controls). Only authorized personnel will be allowed access to them (personnel controls), and even once physical access has been allowed, logical authentication methods (e.g. passwords and/or smartcards) will need to be used (logical controls).

The other two measures that are relevant are that sensitive workloads will need to be placed on these restricted hosts (workload placement) and that secured communications are likely to be needed for various of these components (secure communications) - whether for control plane or data plane.

Given that compromises to the hosted applications may still lead to compromise of the hosting service - and other hosted applications on the same hosting service - measured and secured boot, combined with attestation, are best practice, and any run-time checks that may be available should also be considered.

8.2 Moving to pooled, restricted hosts

The key technical difference between this approach and the previous is the introduction of trusted pools of hosts. Though these may not have a higher security profile than the hosts in outlined in clause 8.1, and still need to be managed with physical, personnel and logical controls, the addition of attestation as a key measure (rather than an optional measure) allows for more sophisticated placement scenarios than is the case with single, restricted hosts.

The addition of the ability to conceal resource usage (as described in clause 7.2.2.2) is relevant only a small set of use cases, but allows for greater integration of Management and Orchestration components with the hosts in the pool. There may be a temptation, given the small number of use cases where resource usage concealment is required (typically only Lawful Interception, as described in clause 4.5), to place the majority of components requiring secure administration into one or more different pools, and to separate out those components associated with Lawful Interception into a separate pool. There are two major drawbacks to this approach however:

1) some of the sensitive components may require siting alongside Lawful Interception components;

2) the more pools that are created, the higher the administrative overhead.

In summary, though pools of restricted hosts offer, at first glance, the appearance of greater administrative control over placement of sensitive components, the trade-offs that may be required need to be carefully weighed against this possible benefit.

8.3 Moving to pooled, unrestricted hosts

The key difference between this approach and the previous is the introduction of hardware-mediated execution enclaves. Associated with this is the ability to provide for resource usage concealment, the implementation of which is expected to be less complex than in use cases where no hardware-mediated execution environment is available. There is not necessarily a requirement to provide the same level of protection for all hosts, and different pools, with different capabilities, may be made available for different workloads, but placement of certain workloads will still require placement close to other workloads - for instance Lawful Interception components - and it may prove simpler to provide the same capability profile for all hosts which will be hosting sensitive components. Of all the approaches, this provides the most flexibility for different deployment architectures.

The addition of hardware-mediated execution enclaves also removes the requirement for some of the measures in the previous approaches. The key one of these is physical controls and alarms (clause 6.5), as once hosted applications can execute sensitive processes - and maintain sensitive data - within a hardware mediated execution enclave, physical access to the hosting system - with the concomitant expectation that root access can be gained with it - is less important, as the system as a whole is less vulnerable to root-based attacks. The same goes for OS-level access control, as processes and data can be better protected whatever the level of access control available to users.

Annex A:
Change history

Date	Version	Information about changes
2024-08	V1.2.2	Initial draft version for ed131 created from published version v1.2.1
2024-09	V1.2.3	Implementation of the following contributions agreed during SEC270 and SEC271 – NFVSEC(24)000171r1_CR_to_SEC009_on_replacing_TPM_with_generic_descriptions – NFVSEC(24)000173r1_SEC009_vHSM_definition – Additional changes in the abbreviations section vHSM virtual HSM VIM Virtual Infrastructure Manager

History

Document history		
V1.1.1	December 2015	Publication as ETSI GS NFV-SEC 009
V1.2.1	January 2017	Publication
V1.3.1	January 2025	Publication