VNF
Cloud Readiness Requirements
for
OpenECOMP

Revision **1.0**

Revision Date **2/1/2017**

Document Revision History

Date	Revision	Description
2/1/2017	1.0	Initial public release of VNF Cloud Readiness Requirements for OpenECOMP

Table of Contents

Definitions

Throughout the document the terms have the following meaning:

MUST This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.

MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.

SHOULD This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.

SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behavior described with this label.

MAY This word, or the adjective "OPTIONAL", mean that an item is truly optional. One vendor may choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

1. Introduction

This document is part of a hierarchy of documents that describes the overall Requirements and Guidelines for OpenECOMP. The diagram below identifies where this document fits in the hierarchy.

<table>
<tr><td colspan="5">OpenECOMP Requirements and Guidelines</td></tr>
<tr><td colspan="4">VNF Guidelines for Network Cloud and OpenECOMP</td><td>Future OpenECOMP Subject Documents</td></tr>
<tr><td>VNF Cloud Readiness Requirements for OpenECOMP</td><td>VNF Management Requirements for OpenECOMP</td><td>VNF Heat Template Requirements for OpenECOMP</td><td>Future VNF Requirements Documents</td><td>Future Requirements Documents</td></tr>
</table>

Document summary:

VNF Guidelines for Network Cloud and OpenECOMP

- Describes VNF environment and overview of requirements

VNF Cloud Readiness Requirements for OpenECOMP

- Cloud readiness requirements for VNFs (Design, Resiliency, Security, and DevOps)

VNF Management Requirements for OpenECOMP

- Requirements for how VNFs interact and utilize OpenECOMP

VNF Heat Template Requirements for OpenECOMP

- Provides recommendations and standards for building Heat templates compatible with OpenECOMP– initial implementations of Network Cloud are assumed to be OpenStack based.

This reference document lists the requirements that are the supporting details for the Virtual Network Function (VNF) characteristics outlined in the *VNF Guidelines for Network Cloud and OpenECOMP*. These requirements are grouped into the following categories: VNF Design, Resiliency, Security, and DevOps. Specific requirements for OpenECOMP can be found in the *VNF Management Requirements for OpenECOMP* reference document.

This section outlines the guidelines for VNFs to be compliant with running on a multi-tenant, Network Cloud infrastructure. VNFs must be virtualized, software-based, execute in a multi-tenant cloud, and be de-coupled from the cloud hardware. To achieve interoperability between VNFs, open and standard interfaces and APIs must be used. The set of reusable VNFs forms the basis of a VNF catalog that is made available to service designers to compose new (service chained) services that can include service-specific custom parameters and QoS policies. Use of open source technologies to leverage industry innovation is important in the design of virtualized services. Equally important is the re-use of common technologies (e.g., virtualized load balancers, firewalls, etc.) that are provided by the platform.

2. VNF Design

Services are composed of VNFs and common components and are designed to be agnostic of the location to leverage capacity where it exists in the Network Cloud. VNFs can be instantiated in any location that meets the performance and latency requirements of the service.

A key design principle for virtualizing services is decomposition of network functions using NFV concepts into granular VNFs. This enables instantiating and customizing only essential functions as needed for the service, thereby making service delivery more nimble. It provides flexibility of sizing and scaling and also provides flexibility with packaging and deploying VNFs as needed for the service. It enables grouping functions in a common cloud data center to minimize inter-component latency. The VNFs should be designed with a goal of being modular and reusable to enable using best-in-breed vendors

Section 4.1.1 in *VNF Guidelines for Network Cloud and OpenECOMP* describes the overall guidelines for designing VNFs from VNF Components (VNFCs). Below are more detailed requirements for composing VNFs.

VNF Design Requirements	Type	ID #
Decompose VNFs into granular re-usable VNFCs	Should	20010
Decompose if the functions have significantly different scaling characteristics (e.g., signaling versus media functions, control versus data plane functions).	Must	20020
Decomposition of the VNF must enable instantiating only the functionality that is needed for the VNF (e.g., if transcoding is not needed it should not be instantiated).	Must	20030
Design VNFC as a standalone, executable process.	Must	20040
Create a single component VNF for VNFCs that can be used by other VNFs.	Should	20050
Design to scale horizontally (more instances of a VNF or VNFC) and not vertically (moving the existing instances to larger VMs or increasing the resources within a VM) to achieve effective utilization of cloud resources.	Must	20060
Utilize cloud provided infrastructure and VNFs (e.g., virtualized Local Load Balancer) as part of the VNF so that the cloud can manage and provide a consistent service resiliency and methods across all VNF's.	Must	20070
VNFCs should be independently deployed, configured, upgraded, scaled, monitored, and administered by OpenECOMP.	Should	20080
Provide API versioning to allow for independent upgrades of VNFC.	Must	20090
Minimize the use of state within a VNFC to facilitate the movement of traffic from one instance to another.	Should	20100
Maintain state in a geographically redundant datastore that may, in fact, be its own VNFC.	Should	20110
Decouple persistent data from the VNFC and keep it in its own datastore that can be reached by all instances of the VNFC requiring the data.	Should	20120
Utilize virtualized, scalable open source database software that can meet the performance/latency requirements of the service for all datastores.	Must	20130
Failure of a VNFC instance must not terminate stable sessions.	Must	20140
Enable DPDK in the guest OS for VNF's requiring high packets/sec performance. High packet throughput is defined as greater than 500K packets/sec.	Must	20150
When using DPDK, use the NCSP's supported library and compute flavor that supports DPDK to optimize network efficiency.[1]	Must	20160
Do not use technologies that bypass virtualization layers (such as SR-IOV) unless approved by the NCSP (e.g., if necessary to meet functional or performance requirements).	Must	20170

[1] Refer to NCSP's Network Cloud specification

VNF Design Requirements	Type	ID #
Limit the size of application data packets to no larger than 907400 bytes for SDN network-based tunneling when guest data packets are transported between tunnel endpoints that support guest logical networks.	Must	20180
Do not require the use of a dynamic routing protocol unless necessary to meet functional requirements.	Must	20190

3. Resiliency

The VNF is responsible for meeting its resiliency goals and must factor in expected availability of the targeted virtualization environment. This is likely to be much lower than found in a traditional data center. Resiliency is defined as the ability of the VNF to respond to error conditions and continue to provide the service intended. A number of software resiliency dimensions have been identified as areas that should be addressed to increase resiliency. As VNFs are deployed into the Network Cloud, resiliency must be designed into the VNF software to provide high availability versus relying on the Network Cloud to achieve that end.

Section 4.1.2 in *VNF Guidelines for Network Cloud and OpenECOMP* describes the overall guidelines for designing VNFs to meet resiliency goals. Below are more detailed resiliency requirements for VNFs.

3.1 All Layer Redundancy

Design the VNF to be resilient to the failures of the underlying virtualized infrastructure (Network Cloud). VNF design considerations would include techniques such as multiple vLANs, multiple local and geographic instances, multiple local and geographic data replication, and virtualized services such as Load Balancers.

All Layer Redundancy Requirements	Type	ID #
VNFs are responsible to meet their own resiliency goals and not rely on the Network Cloud.	Must	30010
Design resiliency into a VNF such that the resiliency deployment model (e.g., active-active) can be chosen at run-time.	Must	30020
VNFs must survive any single points of failure within the Network Cloud (e.g., virtual NIC, VM, disk failure).	Must	30030
VNFs must survive any single points of software failure internal to the VNF (e.g., in memory structures, JMS message queues).	Must	30040
Design, build and package VNFs to enable deployment across multiple fault zones (e.g., VNFCs deployed in different servers, racks, OpenStack regions, geographies) to increase the overall resiliency of the VNF.	Must	30050
Support the ability to failover a VNFC automatically to other geographically redundant sites if not deployed active-active to increase the overall resiliency of the VNF.	Must	30060
Support the ability of the VNFC to be deployable in multi-zoned cloud sites to allow for site support in the event of cloud zone failure or upgrades.	Must	30070

3.2 Minimize Cross Data-Center Traffic

Avoid performance-sapping data center-to-data center replication delay by applying techniques such as caching and persistent transaction paths - Eliminate replication delay impact between data centers by using a concept of stickiness (i.e., once a client is routed to data center "A", the client will stay with Data center “A” until the entire session is completed).

Minimize Cross Data-Center Traffic Requirements	Type	ID #
Minimize the propagation of state information across multiple data centers to avoid cross data center traffic.	Should	31010

3.3 Application Resilient Error Handling

Ensure an application communicating with a downstream peer is equipped to intelligently handle all error conditions. Make sure code can handle exceptions seamlessly - implement smart retry logic and implement multi-point entry (multiple data centers) for back-end system applications.

Application Resilient Error Handling Requirements	Type	ID #
Detect connectivity failure for inter VNFC instance and intra/inter VNF and re-establish connectivity automatically to maintain the VNF without manual intervention to provide service continuity.	Must	32010
Handle the restart of a single VNFC instance without requiring all VNFC instances to be restarted.	Must	32020
Handle the start or restart of VNFC instances in any order with each VNFC instance establishing or re-establishing required connections or relationships with other VNFC instances and/or VNFs required to perform the VNF function/role without requiring VNFC instance(s) to be started/restarted in a particular order.	Must	32030
Handle errors and exceptions so that they do not interrupt processing of incoming VNF requests to maintain service continuity.	Must	32040
Provide the ability to modify the number of retries, the time between retries and the behavior/action taken after the retries have been exhausted for exception handling to allow the Network Cloud Service Provider to control that behavior.	Must	32050
Fully exploit exception handling to the extent that resources (e.g., threads and memory) are released when no longer needed regardless of programming language.	Must	32060
Handle replication race conditions both locally and geo-located in the event of a data base instance failure to maintain service continuity.	Must	32070
Automatically retry/resubmit failed requests made by the software to its downstream system to increase the success rate.	Must	32080

3.4 System Resource Optimization

Ensure an application is using appropriate system resources for the task at hand; for example, do not use network or IO operations inside critical sections, which could end up blocking other threads or processes or eating memory if they are unable to complete. Critical sections should only contain memory operation, and should not contain any network or IO operation.

System Resource Optimization Requirements	Type	ID #
Do not execute long running tasks (e.g., IO, database, network operations, service calls) in a critical section of code, so as to minimize blocking of other operations and increase concurrent throughput.	Must	33010
Automatically advertise newly scaled components so there is no manual intervention required.	Must	33020
Utilize FQDNs (and not IP address) for both Service Chaining and scaling.	Must	33030
Deliver any and all functionality from any VNFC in the pool. The VNFC pool member should be transparent to the client. Upstream and downstream clients should only recognize the function being performed, not the member performing it.	Must	33040
Automatically enable/disable added/removed sub-components or component so there is no manual intervention required.	Should	33050
Support the ability to scale down a VNFC pool without jeopardizing active sessions. Ideally, an active session should not be tied to any particular VNFC instance.	Should	33060

Support load balancing and discovery mechanisms in resource pools containing VNFC instances.	Should	33070
Utilize resource pooling (threads, connections, etc.) within the VNF application so that resources are not being created and destroyed resulting in resource management overhead.	Should	33080
Use techniques such as “lazy loading” when initialization includes loading catalogues and/or lists which can grow over time, so that the VNF startup time does not grow at a rate proportional to that of the list.	Should	33090
Release and clear all shared assets (memory, database operations, connections, locks, etc.) as soon as possible, especially before long running sync and asynchronous operations, so as to not prevent use of these assets by other entities.	Should	33100

3.5 Application Configuration Management

Leverage configuration management audit capability to drive conformity to develop gold configurations for technologies like Java, Python, etc.

Application Configuration Management Requirements	Type	ID #
Allow configurations and configuration parameters to be managed under version control to ensure consistent configuration deployment, traceability and rollback.	Must	34010
Allow configurations and configuration parameters to be managed under version control to ensure the ability to rollback to a known valid configuration.	Must	34020
Allow changes of configuration parameters to be consumed by the VNF without requiring the VNF or its sub-components to be bounced so that the VNF availability is not effected.	Must	34030

3.6 Intelligent Transaction Distribution & Management

Leverage Intelligent Load Balancing and redundant components (hardware and modules) for all transactions, such that at any point in the transaction: front end, middleware, back end -- a failure in any one component does not result in a failure of the application or system; i.e., transactions will continue to flow, albeit at a possibly reduced capacity until the failed component restores itself. Create redundancy in all layers (software and hardware) at local and remote data centers; minimizing interdependencies of components (i.e. data replication, deploying non-related elements in the same container).

Intelligent Transaction Distribution & Management Requirements	Type	ID #
Use intelligent routing by having knowledge of multiple downstream/upstream endpoints that are exposed to it, to ensure there is no dependency on external services (such as load balancers) to switch to alternate endpoints.	Should	35010
Use redundant connection pooling to connect to any backend data source that can be switched between pools in an automated/scripted fashion to ensure high availability of the connection to the data source.	Should	35020
Include control loop mechanisms to notify the consumer of the VNF of their exceeding SLA thresholds so the consumer is able to control its load against the VNF.	Should	35030

3.7 Deployment Optimization

Reduce opportunity for failure, by human or by machine, through smarter deployment practices and automation. This can include rolling code deployments, additional testing strategies, and smarter deployment automation (remove the human from the mix).

Deployment Optimization Requirements	Type	ID #
Support at least two major versions of the VNF software and/or sub-components to co-exist within production environments at any time so that upgrades can be applied across multiple systems in a staggered manner.	Must	36010
Support the existence of multiple major/minor versions of the VNF software and/or sub-components and interfaces that support both forward and backward compatibility to be transparent to the Service Provider usage.	Must	36020
Support staggered/rolling deployments between its redundant instances to allow "soak-time/burn in/slow roll" which can enable the support of low traffic loads to validate the deployment prior to supporting full traffic loads.	Must	36030
Support the ability of a requestor of the service to determine the version (and therefore capabilities) of the service so that Network Cloud Service Provider can understand the capabilities of the service.	Must	36040
Test for adherence to the defined performance budgets at each layer, during each delivery cycle with delivered results, so that the performance budget is measured and the code is adjusted to meet performance budget.	Must	36050
Test for adherence to the defined performance budget at each layer, during each delivery cycle so that the performance budget is measured and feedback is provided where the performance budget is not met.	Must	36060
Test for adherence to the defined resiliency rating recommendation at each layer, during each delivery cycle with delivered results, so that the resiliency rating is measured and the code is adjusted to meet software resiliency requirements.	Should	36070
Test for adherence to the defined resiliency rating recommendation at each layer, during each delivery cycle so that the resiliency rating is measured and feedback is provided where software resiliency requirements are not met.	Should	36080

3.8 Monitoring & Dashboard

Promote dashboarding as a tool to monitor and support the general operational health of a system. It is critical to the support of the implementation of many resiliency patterns essential to the maintenance of the system. It can help identify unusual conditions that might indicate failure or the potential for failure. This would contribute to improve Mean Time to Identify (MTTI), Mean Time to Repair (MTTR), and post-incident diagnostics.

Monitoring & Dashboard Requirements	Type	ID #
Provide a method of metrics gathering for each layer's performance to identify/document variances in the allocations so they can be addressed.	Must	37010
Provide unique traceability of a transaction through its life cycle to ensure quick and efficient troubleshooting.	Must	37020
Provide a method of metrics gathering and analysis to evaluate the resiliency of the software from both a granular as well as a holistic standpoint. This includes, but is not limited to thread utilization, errors, timeouts, and retries.	Must	37030
Provide operational instrumentation such as logging, so as to facilitate quick resolution of issues with the VNF to provide service continuity.	Must	37040
Monitor for and alert on (both sender and receiver) errant, running longer than expected and missing file transfers, so as to minimize the impact due to file transfer errors.	Must	37050
Use an appropriately configured logging level that can be changed dynamically, so as to not cause performance degradation of the VNF due to excessive logging.	Should	37060
Utilize Cloud health checks, when available from the Network Cloud, from inside the application through APIs to check the network connectivity, dropped packets rate, injection, and auto failover to alternate sites if needed.	Should	37070

Conduct a resiliency impact assessment for all inter/intra-connectivity points in the VNF to provide an overall resiliency rating for the VNF to be incorporated into the software design and development of the VNF.	Must	37080

4. Security

The objective of this section is to provide the key security requirements that need to be met by VNFs. The security requirements are grouped into five areas as listed below. Other security areas will be addressed in future updates. These security requirements are applicable to all VNFs. Additional security requirements for specific types of VNFs will be applicable and are outside the scope of these general requirements.

Section 4.1.3 in *VNF Guidelines for Network Cloud and OpenECOMP* outlines the five broad security areas for VNFs that are detailed in the following sections:

- **VNF General Security**: This section addresses general security requirements for the VNFs that the vendors will need to address.
- **VNF Identity and Access Management**: This section addresses security requirements with respect to Identity and Access Management as these pertain to generic VNFs.
- **VNF API Security**: This section addresses the generic security requirements associated with APIs. These requirements are applicable to those VNFs that use standard APIs for communication and data exchange.
- **VNF Security Analytics**: This section addresses the security requirements associated with analytics for VNFs that deal with monitoring, data collection and analysis.
- **VNF Data Protection**: This section addresses the security requirements associated with data protection.

4.1 VNF General Security Requirements

This section provides details on the VNF general security requirements on various security areas such as user access control, network security, ACLs, infrastructure security, and vulnerability management. These requirements cover topics associated with compliance, security patching, logging/accounting, authentication, encryption, role-based access control, least privilege access/authorization. The following security requirements need to be met by the solution in a virtual environment:

General Security Requirements	Type	ID #
Integration and operation within a robust security environment is necessary and expected. The security architecture will include one or more of the following: IDAM (Identity and Access Management) for all system and applications access, Code scanning, network vulnerability scans, OS, Database and application patching, malware detection and cleaning, DDOS prevention, network security gateways (internal and external) operating at various layers, host and application based tools for security compliance validation, aggressive security patch application, tightly controlled software distribution and change control processes and other state of the art security solutions. The VNF is expected to function reliably within such an environment and the developer is expected to understand and accommodate such controls and can expected to supply responsive interoperability support and testing throughout the product's lifecycle.	Informational	40010
The VNF must accommodate the security principle of "least privilege" during development, implementation and operation. The importance of "least privilege" cannot be overstated and must be observed in all aspects of VNF development and not limited to security. This is applicable to all sections of this document.	Must	40020
Implement access control list for OA&M services (e.g., restricting access to certain ports or applications).	Must	40030

General Security Requirements		Type	ID #
Implement Data Storage Encryption (database/disk encryption) for Sensitive Personal Information (SPI) and other subscriber identifiable data. Note: subscriber's SPI/data must be encrypted at rest, and other subscriber identifiable data should be encrypted at rest. Other data protection requirements exist and should be well understood by the developer.		Must	40040
Implement a mechanism for automated and frequent "system configuration (automated provisioning / closed loop)" auditing.		Should	40050
Use both network scanning and application scanning security tools on all code, including underlying OS and related configuration. Scan reports shall be provided. Remediation roadmaps shall be made available for any findings.		Should	40060
Perform source code to scanning tools (e.g., Fortify) and provide reports.		Should	40070
Production code shall be distributed from NCSP internal sources only. No production code, libraries, OS images, etc. shall be distributed from publically accessible depots.		Must	40080
Provide all code/configuration files in a "Locked down" or hardened state or with documented recommendations for such hardening. All unnecessary services will be disabled. Vendor default credentials, community strings and other such artifacts will be removed or disclosed so that they can be modified or removed during provisioning.		Must	40090
Support L3 VPNs that enable segregation of traffic by application (dropping packets not belonging to the VPN) (i.e., AVPN, IPSec VPN for Internet routes).		Should	40100
Interoperate with various access control mechanisms for the Network Cloud execution environment (e.g., Hypervisors, containers).		Should	40110
VNF should support the use of virtual trusted platform module, hypervisor security testing and standards scanning tools.		Should	40120
Interoperate with the OpenECOMP (SDN) Controller so that it can dynamically modify the firewall rules, ACL rules, QoS rules, virtual routing and forwarding rules.		Must	40130
Support the ability to work with aliases (e.g., gateways, proxies) to protect and encapsulate resources.		Should	40140
All access to applications (Bearer, signaling and OA&M) will pass through various security tools and platforms from ACLs, stateful firewalls and application layer gateways depending on manner of deployment. The application is expected to function (and in some cases, interwork) with these security tools.		Must	40150
Patch vulnerabilities in VNFs as soon as possible. Patching shall be controlled via change control process with vulnerabilities disclosed along with mitigation recommendations.		Must	40160
Identification, authentication and access control of **customer** or **VNF application users** must be performed by utilizing the NCSP's IDAM API.		Must	40170
Identification, authentication and access control of **OA&M** and other system level functions must use the NCSP's IDAM API or comply with the following is expected.		Must	40180
	Support User-IDs and passwords to uniquely identify the user/application. VNF needs to have appropriate connectors to the Identity, Authentication and Authorization systems that enables access at OS, Database and Application levels as appropriate.	Must	40190
	Provide the ability to support Multi-Factor Authentication (e.g., 1st factor = Software token on device (RSA SecureID); 2nd factor = User Name+Password, etc.) for the users.	Must	40200
	Support Role-Based Access Control to permit/limit the user/application to performing specific activities.	Must	40210
	Support logging via OpenECOMP for a historical view of "who did what and when".	Must	40220

General Security Requirements		Type	ID #
	Encrypt OA&M access (e.g., SSH, SFTP).	Must	40230
	Enforce a configurable maximum number of Login attempts policy for the users. VNF vendor must comply with "terminate idle sessions" policy. Interactive sessions must be terminated, or a secure, locking screensaver must be activated requiring authentication, after a configurable period of inactivity. The system-based inactivity timeout for the enterprise identity and access management system must also be configurable.	Must	40240
	Comply with the NCSP's credential management policy.	Must	40250
	Password expiration must be required at regular configurable intervals.	Must	40260
	Comply with "password complexity" policy. When passwords are used, they shall be complex and shall at least meet the following password construction requirements: • Be a minimum configurable number of characters in length. • Include 3 of the 4 following types of characters: upper-case alphabetic, lower-case alphabetic, numeric, and special. • Not be the same as the UserID with which they are associated or other common strings as specified by the environment. • Not contain repeating or sequential characters or numbers. • Not to use special characters that may have command functions. • New passwords must not contain sequences of three (3) or more characters from the previous password.	Must	40270
	Comply with "password changes (includes default passwords)" policy. Products will support password aging, syntax and other credential management practices on a configurable basis.	Must	40280
	Support use of common third party authentication and authorization tools such as TACACS+, RADIUS.	Must	40290
	Comply with "No Self-Signed Certificates" policy. Self-signed certificates must be used for encryption only, using specified and approved encryption protocols such as LS 1.1 or higher or equivalent security protocols such as IPSec, AES.	Must	40300
	Authenticate system to system communications where one system accesses the resources of another system, and must never conceal individual accountability.	Must	40310

4.2 VNF Identity and Access Management Requirements

The following security requirements for logging, identity, and access management need to be met by the solution in a virtual environment:

Identity and Access Management Requirements	Type	ID #
Access to VNFs will be required at several layers. Hence, VNF vendor needs to be able to host connectors for access to the following layers:		
a. Application	Must	41010
b. OS (Operating System)	Must	41020
c. Database	Must	41030
Manage access to VNF, its OS, or Database by an enterprise access request process.	Must	41040
Comply with the following when persons or non-person entities access VNFs:		
a. Individual Accountability (each person must be assigned a unique ID)	Must	41050
b. Least Privilege (no more privilege than required to perform job functions)	Must	41060
c. Segregation of Duties (access to a single layer and no developer may access production without special oversight)	Must	41070

Identity and Access Management Requirements	Type	ID #
Vendors will not be allowed to access VNFs remotely, e.g., VPN	Must	41080
Vendors accessing VNFs through a client application API must be authorized by the client application owner and the resource owner of the VNF before provisioning authorization through Role Based Access Control (RBAC), Attribute Based Access Control (ABAC), or other policy based mechanism.	Must	41090
Vendor VNF access will be subject to privilege reconciliation tools to prevent access creep and ensure correct enforcement of access policies.	Must	41100
Provide or Support the Identity and Access Management (IDAM) based threat detection data for:		
a. OWASP Top 10	Must	41110
b. Password Attacks	Must	41120
c. Phishing / SMishing	Must	41130
d. Malware (Key Logger)	Must	41140
e. Session Hijacking	Must	41150
f. XSS / CSRF	Must	41160
g. Replay	Must	41170
h. Man in the Middle (MITM)	Must	41180
i. Eavesdropping	Must	41190
Provide Context awareness data (device, location, time, etc.) and be able to integrate with threat detection system.	Must	41200
Where a VNF vendor requires the assumption of permissions, such as root or administrator, the vendor user must first log in under their individual user login ID then switch to the other higher level account; or where the individual user login is infeasible, must login with an account with admin privileges in a way that uniquely identifies the individual performing the function.	Must	41210
Authenticate system to system access and do not conceal a VNF vendor user's individual accountability for transactions.	Must	41220
Warning Notices: A formal statement of resource intent, i.e., a warning notice, must be made visible upon initial access to a VNF vendor user who accesses private internal networks or Company computer resources, e.g., upon initial logon to an internal web site, system or application which requires authentication.	Must	41230
Use access controls for VNFs and their supporting computing systems at all times to restrict access to authorized personnel only, e.g., least privilege. These controls could include the use of system configuration or access control software.	Must	41240
a. Initial and default settings for new user accounts must provide minimum privileges only.	Must	41250
b. Default settings for user access to sensitive commands and data must be denied authorization.	Must	41260
c. Privileged users may be created conforming to approved request, workflow authorization, and authorization provisioning requirements.	Must	41270
d. Commands affecting network services, such as commands relating to VNFs, must have greater restrictions for access and execution, such as up to 3 factors of authentication and restricted authorization.	Must	41280
Encrypt TCP/IP--HTTPS (e.g., TLS v1.2) transmission of data on internal and external networks.	Must	41290
Unnecessary or vulnerable cgi-bin programs must be disabled.	Must	41300
No public or unrestricted access to any data should be provided without the permission of the data owner. All data classification and access controls must be followed.	Must	41310
When in production, vendors or developers must not do the following without authorization of the VNF system owner including:		
a. Install or use systems, tools or utilities capable of capturing or logging data that was not created by them or sent specifically to them;	Must	41320

Identity and Access Management Requirements	Type	ID #
b. Run security testing tools and programs, e.g., password cracker, port scanners, hacking tools.	Must	41330
Authentication credentials must not be included in security audit logs, even if encrypted.	Must	41340
The standard interface for a VNF should be REST APIs exposed to Client Applications for the implementation of OAuth 2.0 Authorization Code Grant and Client Credentials Grant.	Should	41350
Support hosting connectors for OS Level and Application Access.	Should	41360
Support SCEP (Simple Certificate Enrollment Protocol).	Should	41370

4.3 VNF API Security Requirements

This section covers API security requirements when these are used by the VNFs. Key security areas covered in API security are Access Control, Authentication, Passwords, PKI Authentication Alarming, Anomaly Detection, CALEA, Monitoring and Logging, Input Validation, Cryptography, Business continuity, Biometric Authentication, Identification, Confidentiality and Integrity, and Denial of Service.

The solution in a virtual environment needs to meet the following API security requirements:

API Requirements	Type	ID #
Provide a mechanism to restrict access based on the attributes of the VNF and the attributes of the subject.	Must	42010
Integrate with external authentication and authorization services (e.g., IDAM).	Must	42020
Use certificates issued from publicly recognized Certificate Authorities (CA) for the authentication process where PKI-based authentication is used	Must	42030
Validate the CA signature on the certificate, ensure that the date is within the validity period of the certificate, check the Certificate Revocation List (CRL), and recognize the identity represented by the certificate where PKI-based authentication is used.	Must	42040
Protect the confidentiality and integrity of data at rest and in transit from unauthorized access and modification.	Must	42050
Protect against all denial of service attacks, both volumetric and non-volumetric, or integrate with external denial of service protection tools	Must	42060
Implement at minimum the following input validation controls:		
a. Check the size (length) of all input. Do not permit an amount of input so great that it would cause the VNF to fail. Where the input may be a file, the VNF API must enforce a size limit.	Must	42070
b. Do not permit input that contains content or characters inappropriate to the input expected by the design. Inappropriate input, such as SQL insertions, may cause the system to execute undesirable and unauthorized transactions against the database or allow other inappropriate access to the internal network.	Must	42080
c. Validate that any input file has a correct and valid Multipurpose Internet Mail Extensions (MIME) type. Input files should be tested for spoofed MIME types.	Must	42090
Validate input at all layers implementing VNF APIs.	Must	42100
Comply with NIST standards and industry best practices for all implementations of cryptography	Must	42110
Implement all monitoring and logging as described in the Security Analytics section.	Must	42120
Restrict changing the criticality level of a system security alarm to administrator(s).	Must	42130
Monitor API invocation patterns to detect anomalous access patterns that may represent fraudulent access or other types of attacks, or integrate with tools that implement anomaly and abuse detection.	Must	42140
Support requests for information from law enforcement and government agencies.	Must	42150

4.4 VNF Security Analytics Requirements

This section covers VNF security analytics requirements that are mostly applicable to security monitoring. The VNF Security Analytics cover the collection and analysis of data following key areas of security monitoring:

- Anti-virus software
- Logging
- Data capture
- Tasking
- DPI
- API based monitoring
- Detection and notification
- Resource exhaustion detection
- Proactive and scalable monitoring
- Mobility and guest VNF monitoring
- Closed loop monitoring
- Interfaces to management and orchestration
- Malformed packet detections
- Service chaining
- Dynamic security control
- Dynamic load balancing

The following requirements of security monitoring need to be met by the solution in a virtual environment.

Security Analytics Requirements	Type	ID #
Support the following monitoring features by the VNF:		
a. Real-time detection and notification of security events.	Must	43010
b. Integration functionality via API/Syslog/SNMP to other functional modules in the network (e.g., PCRF, PCEF) that enable dynamic security control by blocking the malicious traffic or malicious end users	Must	43020
c. API-based monitoring to take care of the scenarios where the control interfaces are not exposed, or are optimized and proprietary in nature	Must	43030
d. Event logging, formats, and delivery tools to provide the required degree of event data to OpenECOMP	Must	43040
e. Detection of malformed packets due to software misconfiguration or software vulnerability	Must	43050
f. Integrated DPI/monitoring functionality as part of VNFs (e.g., PGW, MME)	Must	43060
g. Alternative monitoring capabilities when VNFs do not expose data or control traffic or use proprietary and optimized protocols for inter VNF communication	Must	43070
h. Proactive monitoring to detect and report the attacks on resources so that the VNFs and associated VMs can be isolated, such as detection techniques for resource exhaustion, namely OS resource attacks, CPU attacks, consumption of kernel memory, local storage attacks.	Must	43080
Coexist and operate normally with commercial anti-virus software which shall produce alarms every time when there is a security incident.	Must	43090
Protect all security audit logs (including API, OS and application-generated logs), security audit software, data, and associated documentation from modification, or unauthorized viewing, by standard OS access control mechanisms, by sending to a remote system, or by encryption.	Must	43100
Log the following events:		

Security Analytics Requirements	Type	ID #
a. Successful and unsuccessful login attempts	Must	43110
b. Logoffs	Must	43120
c. Successful and unsuccessful changes to a privilege level	Must	43130
d. Starting and stopping of security logging	Must	43140
e. Creating, removing, or changing the inherent privilege level of users	Must	43150
f. Connections to a network listener of the resource	Must	43160
Log, at minimum, the following fields (where applicable and technically feasible) in the security audit logs:		
a. Event type	Must	43170
b. Date/time	Must	43180
c. Protocol	Must	43190
d. Service or program used for access	Must	43200
e. Success/failure	Must	43210
f. Login ID	Must	43220
Security audit logs must never contain an authentication credential, e.g., password, even if encrypted.	Must	43230
Detect when the security audit log storage medium is approaching capacity (configurable) and issue an alarm via SMS or equivalent as to allow time for proper actions to be taken to pre-empt loss of audit data.	Must	43240
Support the capability of online storage of security audit logs.	Must	43250
Activate security alarms automatically when the following events, at a minimum, are detected:		
a. Configurable number of consecutive unsuccessful login attempts	Must	43260
b. Successful modification of critical system or application files	Must	43270
c. Unsuccessful attempts to gain permissions or assume the identity of another user	Must	43280
Include, at a minimum, the following fields in the Security alarms (where applicable and technically feasible):		
a. Date	Must	43290
b. Time	Must	43300
c. Service or program used for access	Must	43310
d. Success/failure	Must	43320
e. Login ID	Must	43330
Restrict changing the criticality level of a system security alarm to administrator(s).	Must	43340
Monitor API invocation patterns to detect anomalous access patterns that may represent fraudulent access or other types of attacks, or integrate with tools that implement anomaly and abuse detection.	Must	43350
Support requests for information from law enforcement and government agencies.	Must	43360
Implement “Closed Loop” automatic implementation (without human intervention) for Known Threats with detection rate in low false positives.	Must	43370
Perform data capture for security functions.	Must	43380
Generate security audit logs that must be sent to Security Analytics Tools for analysis.	Must	43390
Provide audit logs that include user ID, dates, times for log-on and log-off, and terminal location at minimum.	Must	43400
Provide security audit logs including records of successful and rejected system access data and other resource access attempts.	Must	43410
Support the storage of security audit logs for agreed period of time for forensic analysis.	Must	43420
Provide the capability of generating security audit logs by interacting with the operating system (OS) as appropriate.	Must	43430

Security Analytics Requirements	Type	ID #
Security logging for VNFs and their OSs must be active from initialization. Audit logging includes automatic routines to maintain activity records and cleanup programs to ensure the integrity of the audit/logging systems.	Must	43440

4.5 VNF Data Protection Requirements

This section covers VNF data protection requirements that are mostly applicable to security monitoring.

Data Protection Requirements	Type	ID #
Provide the capability to restrict read and write access to data.	Must	44010
Provide the capability to restrict access to data to specific users.	Must	44020
Provide the capability to encrypt data in transit on a physical or virtual network.	Must	44030
Provide the capability to encrypt data on non-volatile memory.	Must	44040
Where the encryption of non-transient data is required on a device for which the operating system performs paging to virtual memory, then if possible disable the paging of the data requiring encryption, if not the virtual memory should be encrypted.	Should	44050
Provide the capability to integrate with an external encryption service.	Must	44060
Use industry standard cryptographic algorithms and standard modes of operations when implementing cryptography.	Must	44070
Use commercial algorithms only when there are no applicable US federal standards for specific cryptographic functions, e.g., public key cryptography, message digests.	Should	44080
The SHA, DSS, MD5, SHA-1 and Skipjack algorithms or other compromised encryption must not be used.	Must	44090
Use, whenever possible, standard implementations of security applications, protocols, and format, e.g., S/MIME, TLS, SSH, IPSec, X.509 digital certificates for cryptographic implementations. These implementations must be purchased from reputable vendors and must not be developed in-house.	Must	44100
A VNF must provide the ability to migrate to newer versions of cryptographic algorithms and protocols with no impact.	Must	44110
Use symmetric keys of at least 112 bits in length.	Must	44120
Use asymmetric keys of at least 2048 bits in length.	Must	44130
Use commercial tools that comply with X.509 standards and produce x.509 compliant keys for public/private key generation. Keys must not be generated or derived from predictable functions or values, e.g., values considered predictable include user identity information, time of day, stored/transmitted data.	Must	44140
Provide the capability to configure encryption algorithms or devices so that they comply with the laws of the United States and those of any country in which there are plans to use data encryption.	Must	44150
Provide the capability of using certificates issued from a Certificate Authority not provided by the VNF vendor.	Must	44160
Provide the capability of allowing certificate renewal and revocation.	Must	44170
Provide the capability of testing the validity of a digital certificate by performing the following:		
a. The CA signature on the certificate must be validated	Must	44180
b. The date the certificate is being used must be within the validity period for the certificate	Must	44190
c. The Certificate Revocation List (CRL) for the certificates of that type must be checked to ensure that the certificate has not been revoked	Must	44200
d. The identity represented by the certificate — the "distinguished name" — must be recognized	Must	44210
Provide the capability of encrypting selected data fields stored or bound for security logs.	Must	44220

Data Protection Requirements	Type	ID #
Provide the capability of deleting data stored in the VNF.	Must	44230
Provide the capability to make data available in order to support requests from law enforcement and government agencies as required by legal or regulatory mandates. Capability must be configurable for MOW deployment.	Must	44240

5. DevOps

This section includes guidelines for vendors to ensure that a Network Cloud Service Provider's operations personnel have a common and consistent way to support VNFs and VNFCs.

NCSPs may elect to support standard images to enable compliance with security, audit, regulatory and other needs. As part of the overall VNF software bundle, VNF suppliers using standard images would typically provide the NCSP with an install package consistent with the default OS package manager (e.g. aptitude for Ubuntu, yum for Redhat/CentOS).

Section 4.1.4 in *VNF Guidelines for Network Cloud and OpenECOMP* describes the DevOps guidelines for VNFs.

Additional requirements will be included in the next release of the document.

DevOps Requirements	Type	ID #
Utilize only the Guest OS versions that are supported by the NCSP's Network Cloud.[2]	Must	50010
Utilize only NCSP supported Guest OS images.[2]	Should	50020
Utilize only NCSP standard compute flavors.[2]	Must	50030
Running VMs will not be backed up in the Network Cloud infrastructure. Bringing a VM back up with the configuration required must be accomplished by using appropriate snapshot images or using persistent storage.	Must	50040
Install VNFC(s) on non-root file systems, unless software is specifically included with the operating system distribution of the guest image.	Must	50050

[2] Refer to NCSP's Network Cloud specification