VMware Avi Load Balancer, Version 30.2.3 Security Target

Version 1.2
May 1st, 2025

Broadcom
3401 Hillview Avenue, Palo Alto, CA 94304

2400 Research Blvd
Suite 395
Rockville, MD 20850

Contents

1 Introduction

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

1.1 Security Target and TOE Reference

This section provides the information needed to identify and control the TOE and the ST.

Table 1 – TOE/ST Identification

Category	Identifier
ST Title	VMware Avi Load Balancer, Version 30.2.3 Security Target
ST Version	1.2
ST Date	May 1, 2025
ST Author	Intertek Acumen Security
TOE Identifier	VMware Avi Load Balancer
TOE Version	30.2.3
TOE Developer	Broadcom
Key Words	Network Device, Load Balancer

1.2 TOE Overview

The TOE is VMware Avi Load Balancer Version 30.2.3 which is a network device running as VMware ESXi based Virtual Machine. The TOE is a distributed software TOE consisting of the VMware Avi Controller (hereafter referred to as Controller) and the VMware Avi Service Engine (hereafter referred to as SE). The software-defined, scale-out architecture provides on-demand autoscaling of elastic load balancers. The distributed software load balancers and the backend applications can scale up or down in response to real-time traffic monitoring. Application load balancing becomes more adaptable and intelligent.

1.3 TOE Description

This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references.

The TOE is a distributed virtual TOE comprised of two components:

1. Avi controller (Controller) – The Controller is the “brain” of the entire system and acts as a single point of intelligence, management, and control across a distributed fabric of enterprise-grade load balancers. The Controller is a virtual machine based on Ubuntu Server 20.04 running on a VMware ESXi 7.0.3 hypervisor with an Intel Xeon Gold 6126 processor.
2. Service Engine (SE) – The SE is the distributed data plane. The SE is a virtual machine based on Ubuntu Server 20.04 running on a VMware ESXi 7.0.3 hypervisor with an Intel Xeon Gold 6126 processor.

The following figure represents a sample TOE deployment:

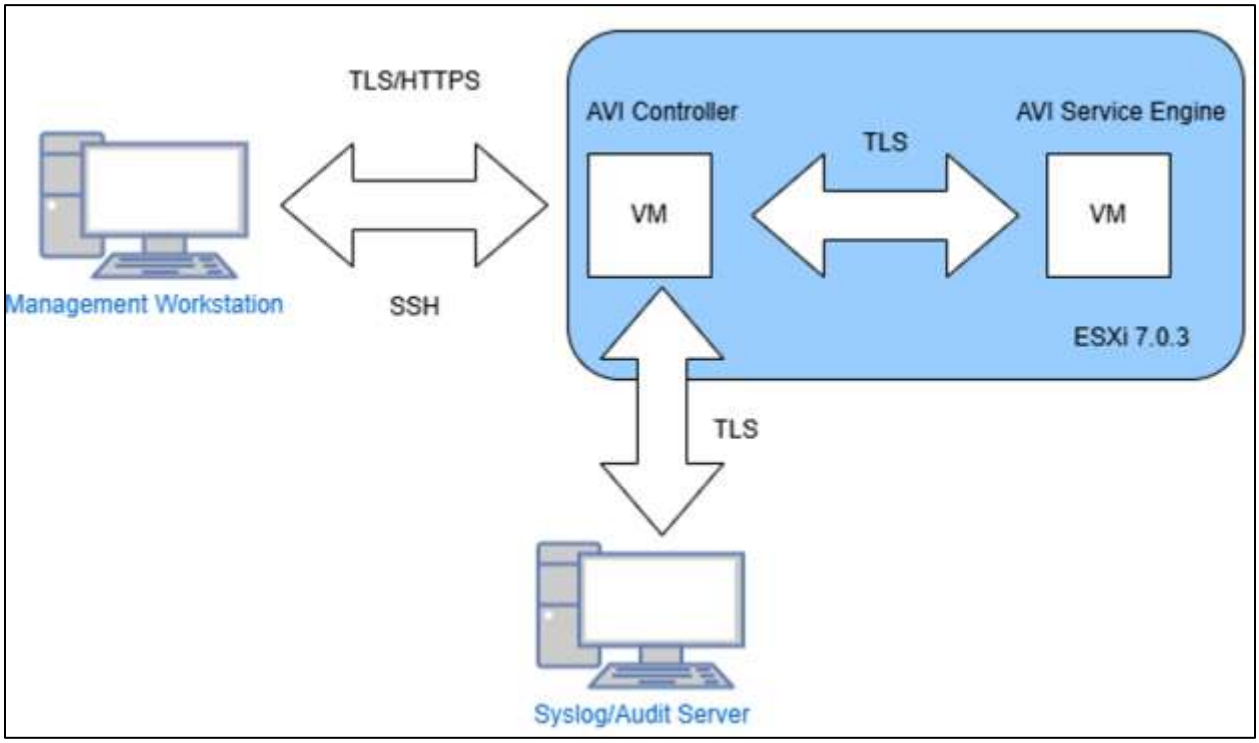

Figure 1 – Representative TOE Deployment

1.3.1 Physical Boundaries

The physical boundaries of the distributed TOE components are described as follows:

- Avi Controller, virtual machine deployed on ESXi 7.0.3
- Avi Service Engine, virtual machine deployed on ESXi 7.0.3

The Controller and SE components are vNDs as defined in Case 1 of NDcPP. The ESXi 7.0.3 hypervisor and underlying hardware platform are part of the evaluated configuration but not included in the TOE boundary.

1.3.2 Security Functions Provided by the TOE

The TOE provides the security functions required by the Collaborative Protection Profile for Network Devices, hereafter referred to as NDcPP v2.2e or NDcPP.

1.3.2.1 Security Audit

The Controller is capable of generating audit records and maintaining a local storage which is rotated when the buffer becomes full. The SE component generated audit records and maintains them temporarily in a local buffer until it has been transferred to the Controller. The Controller component sends its audit records directly to an external syslog server over a trusted channel protected with TLS. The SE component sends its audit records to the Controller component over TLS.

1.3.2.2 Cryptographic Support

The Controller and the SE performs cryptographic functions including key generation and key establishment, symmetric encryption and decryption, hashing, keyed hash message authentication, digital signatures, and random number generation. The following cryptographic libraries is used in support of this functionality:

- VMware's OpenSSL FIPS Object Module

These cryptographic modules were validated on Ubuntu Server 20.04 with ESXi v7.0.3 on an Intel Xeon Gold 6126 processor. The CAVP algorithm certificate details are provided in Section 6.1.
The Controller and the SE cryptographic functionality is implemented in support of TLS v1.2 server (HTTPS) and client (Syslog over TLS) functionality, as necessary to support trusted path and channel functions. Additionally, Controller provides SSHv2 server functionality for remote administration necessary to support trusted path and serves as a trusted channel for manual update functions.

1.3.2.3 Identification and Authentication

Remote and local administrators are authenticated via the Controller component. Repeated failed remote authentications will lock the administrative account after a configurable threshold of attempts. Locked accounts are re-enabled after a configurable time period or can be re-enabled by another administrator. Passwords must meet a configurable minimum length and may be composed of upper-case and lower-case letters, numbers, and special characters. No functionality is available prior to successful authentication. Password characters are obscured during entry.

The Controller and SE components support X.509v3 certificate authentication and revocation checking for the following purposes:

- Validation of a syslog server TLS certificate (Controller).

The Controller is capable of generating certificate signing requests in support of authentication in the following scenarios:

- The Controller TLS server authentication for remote administration and internal distributed TOE communication.
- The SE component also validates the Controller TLS server certificate used for internal distributed TOE channel. Certificate revocation is not supported for the distributed TOE channel.

1.3.2.4 Security Management

The Controller is the primary management component of the TOE and supports local management via VMware console connection as well as remote management via an HTTPS/TLS Web UI and SSH CLI.

The following functionality is available and restricted to authorized security administrators:

- Administer the TOE locally and remotely.
- Configuration of the access banner.
- Configuration of the session inactivity timeouts.
- Performing manual TOE updates.
- Configuring the authentication failure parameters.
- Start and stop services.
- Configuring the transmission of audit data to an external server.
- Management of cryptographic keys.
- Configure the cryptographic functionality.
- Configuring the communication between the Controller and the SE.

- Setting the system time.
- Configure NTP.
- Manage the TOE's trust store and designate X509.v3 certificates as trust anchors.
- Import X.509v3 certificates to the TOE's trust store.
- Manage the trusted public keys database.

1.3.2.5 Protection of the TSF

Administrative passwords are stored in the filesystem of the Controller and are protected via a salted hash. Private keys are stored in the Controller filesystems and public keys are stored in the SE and neither is readable through any TOE interface.

The Controller and SE communicate via an internal trusted channel using TLS.

The system clocks of the Controller and SE components are set manually by the security administrator in support of reliable time stamps. The security administrator can also synchronise time with an NTP server.

The Controller and SE components perform a suite of cryptographic known algorithm tests, entropy noise source tests, and hash signature-based integrity test of the TOE executable code during startup. In the event of a failure of any of the required self-tests, the TOE will shut down its operations until the error can be recovered.

Updates are verified and installed manually by the TOE security administrator using a published hash value.

1.3.2.6 TOE Access

The Controller terminates local administrative sessions after a configurable time period of inactivity. The Controller terminates a remote administrative session after a configurable time period of inactivity. Administrators may terminate their own session by issuing a ‘logout’ command from either the remote CLI or GUI.

Prior to authenticating to the local CLI or the remote CLI or GUI, the administrator is presented with a configurable advisory notice and consent message.

1.3.2.7 Trusted Path/Channels

The TOE acts as a server for the following communications:

- HTTPS server (Controller remote administration)
- SSH server (Controller CLI remote administration)
- TLS server (Controller to SE trusted channel)

The TOE acts as a client for the following communications:

- TLS client (Controller to syslog)
- TLS client (SE to Controller trusted channel)

1.3.3 TOE Documentation

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:

- VMware Avi Load Balancer Security Target, Version 1.1 [ST]
- VMware Avi Load Balancer Administration Manual for Common Criteria, Version 0.5 [AGD]

1.4 TOE Environment

The following environmental components are required to operate the TOE in the evaluated configuration:

Table 2 – TOE Required Environmental Components

Components	Required	Purpose/Description
Syslog server	Required in support of FAU_STG_EXT.1	External syslog server to which audit logs generated on the TOE are sent.
Management workstation	Required in support of FTP_TRP.1/Admin, FMT_SMF.1, FPT_TUD_EXT.1	Administrator workstation from which the TOE is managed remotely and where TOE updates are uploaded from. Local console is achieved via VMware remote administration of the Controller and SE virtual appliances respectively.
CA Server	Required in support of FIA_X509_EXT.1/ITT X.509 Certificate Validation, FIA_X509_EXT.2/ITT X.509 Certificate Validation	The Certificate Authority (CA) server is responsible for issuing, generating, and managing X.509 certificates used with the TOE, as well as handling certificate revocation.
NTP Server	Required in support of FCS_NTP_EXT.1 NTP Protocol	The Controller component implements NTP v4 for reliable time stamps in accordance with RFC 5905.

1.5 Product Functionality not Included in the Scope of the Evaluation

The following product functionality is not included in the CC evaluation:

- High Availability
- Load Balancing
- Orchestrator
- Automation
- Analytics
- Scaling

2 Conformance Claims

This section identifies the TOE conformance claims, conformance rationale, and relevant Technical Decisions (TDs).

2.1 CC Conformance Claims

The TOE is conformant to the following:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017 (Extended)
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017 (Conformant)

2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- Collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020 [CPP_ND_V2.2E]

2.3 Conformance Rationale

This ST provides exact conformance to the items listed in the previous section. The security problem definition, security objectives, and security requirements in this ST are all taken from the NDcPP v2.2e, performing only the operations defined there.

2.3.1 Technical Decisions

All NIAP TDs issued to date and applicable to NDcPP v2.2e have been considered. Table 3 identifies all applicable TDs.

Table 3 – TOE Relevant Technical Decisions

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0527: Updated to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	Yes	
TD0536: NIT Technical Decision for Update Verification Inconsistency	Yes	
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	No	SFR not claimed, The TOE does not implement mutual authentication as TLS client.
TD0546: NIT Technical Decision for DTLS – clarification of Application Note 63	No	The TOE does not implement DTLS.

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	Yes	
TD0556: NIT Technical Decisions for RFC 5077 question	Yes	
TD0563: NIT Technical Decision for Clarification of audit date information	Yes	
TD0564: NIT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	SFR not claimed, The TOE does not implement DTLS.
TD0570: NIT Technical Decision for Clarification about FIA_AFL.1	Yes	
TD0571: NIT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
TD0572: NIT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	Yes	

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0592: NIT Technical Decision for Local Storage of Audit Records	Yes	
TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server	Yes	
TD0632: NIT Technical Decision for Consistency with Time Data for vNDs	Yes	
TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters	Yes	
TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	SFR not claimed, TOE does not act as a SSH client.
TD0638: NIT Technical Decision for Key Pair Generation for Authentication	Yes	
TD0639: NIT Technical Decision for Clarification for NTP MAC Keys	Yes	
TD0670: NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	Yes	
TD0738: NIT Technical Decision for Link to Allowed-With List	Yes	
TD0790: NIT Technical Decision: Clarification Required for testing IPv6	Yes	
TD0792: NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes	
TD0800: Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	No	The TOE does not implement IPsec.

3 Security Problem Definition

The security problem definition has been taken directly from the claimed PP and any relevant EPs/Modules/Packages specified in Section 2.2 and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any Organizational Security Policies (OSPs) that the TOE is expected to enforce.

3.1 Threats

The threats included in Table 4 are drawn directly from the PP and any EPs/Modules/Packages specified in Section 2.2.

Table 4 – Threats

ID	Threat
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate

ID	Threat
	the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic and may allow them to take advantage of any trust relationships with other Network Devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access,

ID	Threat
	change or modify device data, critical network traffic or security functionality of the device.

3.2 Assumptions

The assumptions included in Table 5 are drawn directly from PP and any relevant EPs/Modules/Packages.

Table 5 – Assumptions

ID	Assumption
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality). If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.

ID	Assumption
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.

ID	Assumption
A.COMPONENTS_RUNNING (applies to distributed TOEs only)	For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.
A.VS_TRUSTED_ADMINISTRATOR	The Security Administrators for the VS are assumed to be trusted and to act in the best interest of security for the organization. This includes not interfering with the correct operation of the device. The Network Device is not expected to be capable of defending against a malicious VS Administrator that actively works to bypass or compromise the security of the device.
A.VS_REGULAR_UPDATES	The VS software is assumed to be updated by the VS Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.VS_ISOLATION (applies to vNDs only)	For vNDs, it is assumed that the VS provides, and is configured to provide sufficient isolation between software running in VMs on the same physical platform. Furthermore, it is assumed that the VS adequately protects itself from software running inside VMs on the same physical platform.
A.VS_CORRECT_CONFIGURATION	For vNDs, it is assumed that the VS and VMs are correctly configured to support ND functionality implemented in VMs.

3.3 Organizational Security Policies

The OSPs included in **Table** 6 – OSPs

are drawn directly from the PP and any relevant EPs/Modules/Packages.Table 6 – OSPs

ID	OSP
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

4 Security Objectives

The security objectives have been taken directly from the claimed PP and any relevant EPs/Modules/Packages and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the Operational Environment

Security objectives for the operational environment assist the TOE in correctly providing its security functionality. These objectives, which are found in the Table 7 below, track with the assumptions about the TOE operational environment.

Table 7 – Security Objectives for the Operational Environment

ID	Objectives for the Operational Environment
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

ID	Objectives for the Operational Environment
OE.COMPONENTS_RUNNING (applies to distributed TOEs only)	For distributed TOEs, the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.
OE.VM_CONFIGURATION	For vNDs, the Security Administrator ensures that the VS and VMs are configured to • Reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and • Correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting).

5 Security Requirements

This section identifies the Security Functional Requirements (SFRs) for the TOE. The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revisions 5, April 2017, and all international interpretations.

Table 8 – SFRs

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_GEN_EXT.1	Security Audit Data Generation for Distributed TOE Component
FAU_STG.1	Protected Audit Trail Storage
FAU_STG_EXT.1	Protected Audit Event Storage
FAU_STG_EXT.4	Protected Local Audit Event Storage for Distributed TOEs
FAU_STG_EXT.5	Protected Remote Audit Event Storage for Distributed TOEs
FCO_CPC_EXT.1	Component Registration Channel Definition
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_HTTPS_EXT.1	HTTPS Protocol
FCS_NTP_EXT.1	NTP Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHS_EXT.1	SSH Server Protocol
FCS_TLSC_EXT.1	TLS Client Protocol without Mutual Authentication
FCS_TLSS_EXT.1	TLS Server Protocol
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.1/ITT	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests

Requirement	Description
FMT_MOF.1/Functions	Management of Security Functions Behaviour
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
FMT_MTD.1/Services	Management of Security Functions Behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on security roles
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF Testing
FPT_ITT.1 FPT_ITT.1/Join	Basic internal TSF data transfer protection
FPT_STM_EXT.1	Reliable Time Stamps
FPT_TUD_EXT.1	Trusted Update
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_TAB.1	Default TOE Access Banner
FTP_ITC.1	Inter-TSF Trusted Channel
FTP_TRP.1/Admin	Trusted Path

5.1 Conventions

The CC allows the following types of operations to be performed on the functional requirements: assignments, selections, refinements, and iterations. The following font conventions are used within this document to identify operations defined by CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold text** and ~~strikethroughs~~;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operations were completed in the PP and relevant EPs/Modules/Packages, the formatting used in the PP has been retained.
- Extended SFRs are identified by the addition of “EXT” after the requirement name.

5.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1
The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) *All administrative actions comprising:*
   - *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
   - *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
   - *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
   - *Resetting passwords (name of related user account shall be logged).*
   - *[no other actions];*
d) *Specifically defined auditable events listed in* Table 9

FAU_GEN.1.2
The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of Table 9*.

Table 9 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_GEN_EXT.1	None	None
FAU_STG.1	None	None
FAU_STG_EXT.1	None	None
FAU_STG_EXT.4	None	None
FAU_STG_EXT.5	None	None
FCO_CPC_EXT.1	• Enabling communications between a pair of components • Disabling communications between a pair of components	• Identities of the endpoint pairs enabled or disabled
FCS_CKM.1	None	None

Requirement	Auditable Events	Additional Audit Record Contents
FCS_CKM.2	None	None
FCS_CKM.4	None	None
FCS_COP.1/DataEncryption	None	None
FCS_COP.1/SigGen	None	None
FCS_COP.1/Hash	None	None
FCS_COP.1/KeyedHash	None	None
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure
FCS_NTP_EXT.1	• Configuration of a new time server • Removal of configured time server	• Identity if new/removed time server
FCS_RBG_EXT.1	None	None
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded	Origin of the attempt (e.g., IP address)
FIA_PMG_EXT.1	None	None
FIA_UIA_EXT.1	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UAU_EXT.2	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UAU.7	None	None
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.1/ITT	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store

Requirement	Auditable Events	Additional Audit Record Contents
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None	None
FMT_MOF.1/Functions	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None
FMT_MOF.1/Services	None	None
FMT_MTD.1/CoreData	None	None
FMT_MTD.1/CryptoKeys	None	None
FMT_SMF.1	All management activities of TSF data	None
FMT_SMR.2	None	None
FPT_SKP_EXT.1	None	None
FPT_APW_EXT.1	None	None
FPT_ITT.1 FPT_ITT.1/Join	• Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel functions	Identification of the initiator and target of failed trusted channels establishment attempt
FPT_TST_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None
FTA_SSL.3	The termination of a remote session by the session locking mechanism	None
FTA_SSL.4	The termination of an interactive session	None
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FTA_TAB.1	None	None
FTP_ITC.1	• Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel functions	Identification of the initiator and target of failed trusted channels establishment attempt
FTP_TRP.1/Admin	• Initiation of the trusted path • Termination of the trusted path. • Failure of the trusted path functions.	None

5.2.1.2 FAU_GEN.2 User Identity Association

FAU_GEN.2.1
For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.3 FAU_GEN_EXT.1 Security Audit Data Generation for Distributed TOE Component

FAU_GEN_EXT.1.1
The TSF shall be able to generate audit records for each TOE component. The audit records generated by the TSF of each TOE component shall include the subset of security relevant audit events which can occur on the TOE component.

5.2.1.4 FAU_STG.1 Protected Audit Trail Storage

FAU_STG.1.1
The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2
The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

5.2.1.5 FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.1
The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2
The TSF Shall be able to store generated audit data on the TOE itself. In addition [

- *The TOE shall be a distributed TOE that stores audit data on the following TOE components: [Controller],*

- *The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [SE transmits its audit events to the Controller via TLS as defined in FPT_ITT.1]*

].

FAU_STG_EXT.1.3
The TSF shall [*overwrite previous audit records according to the following rule: [once the active audit buffer reaches a maximum size, a new file is created. Up to 1000 files may be stored; the oldest file will be deleted]*] when the local storage space for audit data is full.

5.2.1.6 FAU_STG_EXT.4 Protected Local Audit Event Storage for Distributed TOEs

FAU_STG_EXT.4.1
The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: *[*

TOE Component	***Action***
Controller	*[overwrite previous audit records according to the following rule [once the active audit buffer reaches a maximum size, a new file is created. Up to 1000 files may be stored; the oldest file will be deleted]]*

].

5.2.1.7 FAU_STG_EXT.5 Protected Remote Audit Event Storage for Distributed TOEs

FAU_STG_EXT.5.1
Each TOE component which does not store security audit data locally shall be able to buffer security audit data locally until it has been transferred to another TOE component that stores or forwards it. All transfer of audit records between TOE components shall use a protected channel according to [*FPT_ITT.1*].

5.2.2 Communication Partner Control (FCO)

5.2.2.1 FCO_CPC_EXT.1 Component Registration Channel Definition

FCO_CPC_EXT.1.1
The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

FCO_CPC_EXT.1.2
The TSF shall implement a registration process in which components establish and use a communications channel that uses [

- *A channel that meets the secure channel requirements in [FPT_ITT.1]*

]
for at least *TSF data*.

Application Note: The secure channel that is used in the registration process has been iterated as FPT_ITT.1/Join. This channel is then subsequently adopted as a continuing internal communication channel between the different TOE components.

FCO_CPC_EXT.1.3
The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.

5.2.3 Cryptographic Support (FCS)

5.2.3.1 FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1
The TSF shall generate **asymmetric** cryptographic key in accordance with a specified cryptographic key generation algorithm: [

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*
- *ECC schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;*
- *FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526,].*

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].~~

5.2.3.2 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1
The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- *RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1";*
- *Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";*
- *FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 3526,].*

] ~~that meets the following: [assignment: list of standards].~~

Application Note: This SFR has been updated as per TD0580 and TD0581

5.2.3.3 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1
The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- *For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of [zeroes]];*

- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [*
    - *logically addresses the storage location of the key and performs a* [*single*] *overwrite consisting of* [*a new value of the key]*]

that meets the following: *No Standard*

5.2.3.4 FCS_COP.1/DataEncryption Cryptographic Operations (AES Data Encryption/Decryption)

FCS_COP.1.1/DataEncryption
The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [*CBC, CTR, GCM*] *mode* and cryptographic key sizes [*128 bits, 256 bits*] that meet the following: *AES as specified in ISO 18033-3,* [*CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772*].

5.2.3.5 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

FCS_COP.1.1/SigGen
The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- *RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits and 3072 bits]*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits, 384 bits and 521 bits]*

]
that meet the following: [

- *For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,*
- *For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4*

].

5.2.3.6 FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)

FCS_COP.1.1/Hash
The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [*SHA-1, SHA-256, SHA-384, SHA-512*] ~~and cryptographic key sizes~~ [~~*assignment: cryptographic key sizes*~~] and **message digest sizes [*160, 256, 384, 512*] bits** that meet the following: *ISO/IEC 10118-3:2004*.

5.2.3.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

FCS_COP.1.1/KeyedHash
The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [*HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512,*] and cryptographic key sizes *[256 bits, 512 bits]* **and message digest sizes [*160, 256, 384, 512*] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.3.8 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1
The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2
The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3
If a peer certificate is presented, the TSF shall [*not require client authentication*] if the peer certificate is deemed invalid.

5.2.3.9 FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1.1
The TSF shall use only the following NTP version(s) [*NTP v4 (RFC 5905)*].

FCS_NTP_EXT.1.2
The TSF shall update its system time using [

- Authentication using [*SHA1*] as the message digest algorithm(s).

]

FCS_NTP_EXT.1.3
The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4
The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

5.2.3.10 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1
The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [*CTR_DRBG (AES)*].

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[one] software-based noise source*] with a minimum of [*256 bits*] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

5.2.3.11 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1
The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [4256, 4344, 5656, 6668, 8268, *8308 section 3.1, 8332].*

FCS_SSHS_EXT.1.2
The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [*password-based*].

Application Note: This SFR has been updated as per TD0636.

FCS_SSHS_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than *[256K]* bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4
The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [*aes128-ctr, aes256-ctr*].

FCS_SSHS_EXT.1.5
The TSF shall ensure that the SSH public-key based authentication implementation uses [*ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp521*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6
The TSF shall ensure that the SSH transport implementation uses [*hmac-sha2-512*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7
The TSF shall ensure that [*ecdh-sha2-nistp256*] and [*diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8
The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

5.2.3.12 FCS_TLSC_EXT.1 TLS Client Protocol without Mutual Authentication

FCS_TLSC_EXT.1.1
The TSF shall implement [*TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

[

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC5246*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC5289*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC5289*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC5289*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC5289*

- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*

] and no other ciphersuites.

FCS_TLSC_EXT.1.2
The TSF shall verify that the presented identifier matches [*the reference identifier per RFC 6125 section 6,] and no other attribute types*].

FCS_TLSC_EXT.1.3
When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [

- *Not implement any administrator override mechanism*].

FCS_TLSC_EXT.1.4
The TSF shall [*present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups:* [*secp256r1, secp384r1, secp521r1*] *and no other curves/groups*] in the Client Hello.

5.2.3.13 FCS_TLSS_EXT.1 TLS Sever Protocol Without Mutual Authentication

FCS_TLSS_EXT.1.1
The TSF shall implement [*TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

[

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC5246*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*

] and no other ciphersuites.

FCS_TLSS_EXT.1.2
The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [*none*].

FCS_TLSS_EXT.1.3
The TSF shall perform key establishment for TLS using [*RSA with key size* [*2048 bits, 3072 bits*], *ECDHE curves* [*secp256r1, secp384r1, secp521r1*] *and no other curves*]].

FCS_TLSS_EXT.1.4
The TSF shall support [*session resumption based on session tickets according to RFC 5077*].

5.2.4 Identification and Authentication (FIA)

5.2.4.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within *[3-20]* unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [*prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed*].

5.2.4.2 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1
The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [*"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [" ", "+"]*]

b) Minimum password length shall be configurable to between [*8*] and [*32*] characters.

5.2.4.3 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1
The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [*no other actions*].

FIA_UIA_EXT.1.2
The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

5.2.4.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1
The TSF shall provide a local [*password-based, SSH public key-based*] authentication mechanism to perform local administrative user authentication.

5.2.4.5 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1
The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress **at the local console**.

5.2.4.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

FIA_X509_EXT.1.1/Rev
The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates**.

- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [*a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2/Rev
The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.4.7 FIA_X509_EXT.1/ITT X.509 Certificate Validation

FIA_X509_EXT.1.1/ITT
The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of two certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [*no revocation method*]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field*.

FIA_X509_EXT.1.2/ITT
The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.4.8 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [*HTTPS, TLS*] and [*no additional uses*].

FIA_X509_EXT.2.2
When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [*not accept the certificate*].

Application Note: This SFR has been updated as per TD0537.

5.2.4.9 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [*Common Name, Organization, Organizational Unit, Country*].

FIA_X509_EXT.3.2
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.2.5 Security Management (FMT)

5.2.5.1 FMT_MOF.1/Functions Management of Security Functions Behaviour.

FMT_MOF.1.1/Functions
The TSF shall restrict the ability to [*modify the behaviour of*] the functions [*transmission of audit data to an external IT entity*] to *Security Administrators*.

5.2.5.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behavior

FMT_MOF.1.1/ManualUpdate
The TSF shall restrict the ability to enable the functions *to perform manual updates to Security Administrators.*

5.2.5.3 FMT_MOF.1/Services Management of Security Functions Behaviour

FMT_MOF.1.1/Services
The TSF shall restrict the ability to **start and stop** ~~the functions~~ **services** to *Security Administrators*.

5.2.5.4 FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData
The TSF shall restrict the ability to manage the *TSF data to Security Administrators.*

5.2.5.5 FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys
The TSF shall restrict the ability to *manage* the *cryptographic keys* to *Security Administrators*.

5.2.5.6 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1
The TSF shall be capable of performing the following management functions:

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [hash comparison] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [

- *Ability to start and stop services;*
- *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
- *Ability to manage the cryptographic keys;*
- *Ability to configure the cryptographic functionality;*
- *Ability to configure the interaction between TOE components;*
- *Ability to set the time which is used for time-stamps;*
- *Ability to configure NTP;*
- *Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;*
- *Ability to import X.509v3 certificates to the TOE's trust store;*
- *Ability to manage the trusted public keys database;*

].

5.2.5.7 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1
The TSF shall maintain the roles:

- *Security Administrator*

FMT_SMR.2.2
The TSF shall be able to associate users with roles.

FMT_SMR.2.3
The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely;*

are satisfied.

5.2.6 Protection of the TSF (FPT)

5.2.6.1 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1
The TSF shall store administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2
The TSF shall prevent the reading of plaintext administrative passwords.

5.2.6.2 FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1
The TSF shall protect TSF data from disclosure and **detect its** modification when it is transmitted between separate parts of the TOE **through the use of [*TLS*]**.

5.2.6.3 FPT_ITT.1/Join Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1/Join
The TSF shall protect TSF data from disclosure and **detect its** modification when it is transmitted between separate parts of the TOE **through the use of [TLS]**.

Application Note: The secure channel that is used in the registration process has been iterated as FPT_ITT.1/Join. This channel is then subsequently adopted as a continuing internal communication channel between the different TOE components.

5.2.6.4 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric, and private keys)

FPT_SKP_EXT.1.1
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.2.6.5 FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1.1
The TSF shall be able to provide reliable time stamps for its own use.

FPT_STM_EXT.1.2
The TSF shall [*allow the Security Administrator to set the time, synchronise time with an NTP server*].

5.2.6.6 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [*during initial start-up (on power on)*] to demonstrate the correct operation of the TSF: [

- *Software integrity test;*
- *OpenSSL crypto module Known Answer Tests.*

].

5.2.6.7 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1
The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [*no other TOE firmware/software version*].

FPT_TUD_EXT.1.2
The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [*no other update mechanism*].

FPT_TUD_EXT.1.3
The TSF shall provide means to authenticate firmware/software updates to the TOE using a [*published hash*] prior to installing those updates.

5.2.7 TOE Access (FTA)

5.2.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1
The TSF shall, for local interactive sessions, [

- *terminate the session*]

after a Security Administrator-specified time period of inactivity

5.2.7.2 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1
The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity.*

5.2.7.3 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1
The TSF shall allow **Administrator**-initiated termination of the **Administrator's** own interactive session.

5.2.7.4 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1
Before establishing **an administrative user** session the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

5.2.8 Trusted Path/Channels (FTP)

5.2.8.1 FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1

The TSF shall **be capable of using [*TLS*]** to provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [*no other capabilities*]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data.**

FTP_ITC.1.2
The TSF shall permit **the TSF or the authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for *[audit server]*.

5.2.8.2 FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin
The TSF shall **be capable of using [*SSH, TLS, HTTPS*] to** provide a communication path between itself and **authorized** remote **Administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **disclosure and provides detection of modification of the channel data**.

FTP_TRP.1.2/Admin
The TSF shall permit remote **Administrators** to initiate communication via the trusted path.

FTP_TRP.1.3/Admin
The TSF shall require the use of the trusted path for *initial Administrator authentication and all remote administration actions*.

5.3 TOE SFR Dependencies Rationale for SFRs

The PP and any relevant EPs/Modules/Packages contain(s) all the requirements claimed in this ST. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the PP and any relevant EPs/Modules/Packages, which is/are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 10.

Table 10 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic functionality specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative Procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing – conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

5.5 Assurance Measures

The TOE satisfied the identified assurance requirements. This section identifies the Assurance Measures applied by Broadcom to satisfy the assurance requirements. The following Table 11 lists the details.

Table 11 – TOE Security Assurance Measures

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the

SAR Component	How the SAR will be met
	condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_CMS.1	
ATE_IND.1	Vendor will provide the TOE for testing.
AVA_VAN.1	Vendor will provide the TOE for testing. The vendor will provide a document identifying the list of software and hardware components.

6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

Table 12 – TOE Summary Specification SFR Description

<table>
<tr><th>Requirement</th><th>TOE Component</th><th>TSS Description</th></tr>
<tr><td>FAU_GEN.1
FAU_GEN.2
FAU_GEN_EXT.1</td><td>Controller, SE</td><td>The TOE generates audit records for both the Controller and the SE components according to the table below. The audit records contain the date/time at which the event occurred, the subject identity (if applicable), the outcome (success or failure), and the type of event, including the additional audit record content identified in Table 9 above. In addition to the events in the table below, the TOE also audits the startup and shutdown of the audit function, administrator login and logout, password changes, configuration changes, and cryptographic key management operations.
When key management operations occur as part of public key and/or certificate enrolment/renewal, the following parameters are logged to identify the relevant key:
Authentication key ID
Subject key ID
Fingerprints
Hostname
<table>
<tr><th>Requirement</th><th>TOE Component</th><th>Auditable Events</th></tr>
<tr><td>FCO_CPC_EXT.1</td><td>Controller, SE</td><td>Enabling communications between a pair of components
Disabling communications between a pair of components</td></tr>
<tr><td>FCS_HTTPS_EXT.1</td><td>Controller</td><td>Failure to establish a HTTPS Session</td></tr>
<tr><td>FCS_NTP_EXT.1</td><td>Controller</td><td>Configuration of a new time server</td></tr>
</table>
</td></tr>
</table>

				• Removal of configured time server
		FCS_SSHS_EXT.1	Controller	Failure to establish an SSH session
		FCS_TLSC_EXT.1	Controller, SE	Failure to establish a TLS Session
		FCS_TLSS_EXT.1	Controller	Failure to establish a TLS Session
		FIA_AFL.1	Controller	Unsuccessful login attempts limit is met or exceeded
		FIA_UIA_EXT.1	Controller	All use of identification and authentication mechanism
		FIA_UAU_EXT.2	Controller	All use of identification and authentication mechanism
		FIA_X509_EXT.1/Rev	Controller	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store
		FIA_X509_EXT.1/ITT	SE	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store

		FMT_MOF.1/ManualUpdate	Controller, SE	Any attempt to initiate a manual update
		FMT_SMF.1	Controller	All management activities of TSF data
		FPT_ITT.1 FPT_ITT.1/Join	Controller, SE	• Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel functions
		FPT_STM_EXT.1	Controller, SE	Discontinuous changes to time - either Administrator actuated or changed via an automated process (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)
		FPT_TUD_EXT.1	Controller, SE	Initiation of update; result of the update attempt (success or failure)
		FTA_SSL.3	Controller	The termination of a remote session by the session locking mechanism

		FTA_SSL.4	Controller	The termination of an interactive session
		FTA_SSL_EXT.1 (if “lock the session” is selected)	Controller	Any attempts at unlocking of an interactive session
		FTP_ITC.1	Controller	• Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel functions
		FTP_TRP.1/Admin	Controller	• Initiation of the trusted path • Termination of the trusted path. • Failure of the trusted path functions.
FAU_STG.1	Controller	TOE is a distributed TOE consisting of two components that is Controller and a Service Engine. Audit data is stored on the Controller component. For SE, audit logs are transmitted to the Controller via TLS 1.2. The SE maintain a local audit buffer which is contained in various files with a fixed number of bytes for each file. “Audit files can be edited via CLI. Only the administrator can edit the file” SE Log Agent - Logs are by default rotated after 4MB (configureable in se_group_properties with field log_agent_file_sz_appl and log_agent_file_sz_conn Access to the audit data of the TOE and the ability to modify or delete it is restricted to authorized administrators only. 1. **Authorization**: Only designated administrators are permitted to initiate log delete operations. 2. **Retention Period:** Logs may only be deleted after the expiration of the predefined retention period specified by organizational policies or regulatory requirements. This ensures that logs are retained for		

		the necessary duration for auditing, compliance, and debugging purposes. SE Log Agent - Allocation to App logs are determined by knobs log_agent_max_storage_ignore_percent, log_agent_max_storage_excess_percent and log_agent_log_storage_min_sz all part of se_group_properties
FAU_STG_EXT.1 FAU_STG_EXT.4 FAU_STG_EXT.5	Controller, SE	TOE is a distributed TOE consisting of two components that is Controller and a Service Engine. The Controller component utilizes a syslog over TLS 1.2 (RFC 5246) and TLS 1.1 (RFC 4346) connection to support transferring audit data to an external audit server. Audit data is stored on the Controller component. For SE, audit logs are transmitted to the Controller via TLS 1.2, according to FPT_ITT.1. The SE maintains a local audit buffer which is contained in various files with a fixed number of bytes for each file. SE Log Agent - Logs are by default rotated after 4MB (configurable in SE group properties with field log_agent_file_sz_appl and log_agent_file_sz_conn). When the local buffer size is exhausted, a copy of the active buffer is archived to a new file and the active buffer is overwritten with an empty file. The Controller component maintains a maximum of 1000 archive files; once this threshold is reached, the oldest archive is deleted. Audit records are sent from the SE to the Controller in real-time as they are generated. The Controller sends its audit records, as well as the audit records received from the SE to the external audit server in real-time as they are generated/received. Audit records on the Controller are only accessible to the Security Administrator.
FCO_CPC_EXT.1	Controller, SE	The Controller and the SE communicate over TLS. In this case, Controller is the TLS server, and the SE is the TLS client. The Security Administrator uses the Controller component to register the SE endpoint's user and its private and public key. The Security Administrator configures the SE user and its corresponding public key on the SE according to instructions in the VMware Avi Load Balancer Administration Manual for Common Criteria. This enables the SE to establish communication with the Controller and to receive configuration updates. The registration channel between the TOE components has been iterated as FPT_ITT.1/Join, which is subsequently adopted as a continuing internal communication channel. The SE identifies the controller based on the OVA and server certificate of the controller which are pre-built in the OVA.

<table>
<tr><td></td><td></td><td>SE has a unique Token, and certificates are used for joining components. The SE always initiates the connection.
The Security Administrator has to delete the Service Engine manually to disable the SEs from the Controller.</td></tr>
<tr><td>FCS_CKM.1
FCS_CKM.2</td><td>Controller, SE</td><td>The Controller and SE components are capable of generating P-256, P-384, and P-521 ECDSA keypairs in accordance with FIPS PUB 186-4 Appendix B.4, and 2048 and 3072-bits RSA keypairs in accordance with FIPS PUB 186-4 Appendix B.3. All applicable SHOULD and SHALL statements are implemented. Additionally, the Controller component is also capable of generating FFC keys in accordance with SP 800-56A revision 3 and RFC 3526 for Groups 14, 16 and 18 i.e. MODP 2048-bit, 4096-bit and 8192-bit groups respectively.
The Controller and SE components implement ECC key establishment in accordance with SP 800-56A revision 3 for TLS and for SSH. Controller also implements FFC key establishment in accordance with SP 800-56A revision 3 using the “safe-prime groups” as defined in RFC 3526 for the SSH protocol. Additionally, the Controller implements RSA key establishment in accordance with Section 7.2 of RFC 3447 for TLS.
The usage of each key generation and key establishment scheme is highlighted in the table below:
<table>
<tr><th>Scheme</th><th>SFR</th><th>Service</th></tr>
<tr><td>RSA key generation and key establishment.

RSAES-PKCS1</td><td>FCS_SSHS_EXT.1</td><td>SSH Remote Administration</td></tr>
<tr><td>RSA key generation and key establishment.

RSAES-PKCS1</td><td>FCS_TLSC_EXT.1
FCS_TLSS_EXT.1
FIA_X509_EXT.1/Rev
FIA_X509_EXT.2
FIA_X509_EXT.3</td><td>Trusted communication channel between itself and authorized IT entities. (Remote administration and Syslog)</td></tr>
<tr><td>ECC key generation and key establishment</td><td>FCS_TLSC_EXT.1
FCS_TLSS_EXT.1
FCS_SSHS_EXT.1
FIA_X509_EXT.1/Rev
FIA_X509_EXT.2
FIA_X509_EXT.3
FIA_X509_EXT.1/ITT</td><td>Trusted communication channel between controller and authorised IT entities (Remote</td></tr>
</table>
</td></tr>
</table>

				administration and audit server).
		ECC key generation and key establishment	FCS_TLSC_EXT.1 FCS_TLSS_EXT.1 FIA_X509_EXT.1/ITT	Trusted communication between Controller and SE.
		FFC key generation and key establishment	FCS_SSHS_EXT.1	Trusted communications between Controller and External IT entity (remote administration)
FCS_CKM.4	Controller, SE	The cryptographic keys utilized by the TOE, their purpose, storage, and key destruction methods, are defined in Table 15. All keys in the volatile memory are considered in use until the TOE is rebooted. All the keys in the non-volatile storage are considered in use until the TOE is re-installed. There are no delays for the key zeroization and as soon as the sensitive data will no longer be used, it is zeroized immediately. The Security administrator has no role in key destruction, and it done by the TOE on its own.		
FCS_COP.1/DataEncryption	Controller, SE	The TOE supports encryption/decryption with AES 128-bit and 256-bit key sizes as specified in ISO 18033-3, CBC as specified in ISO 10116, GCM as specified in ISO 19772 and CTR as specified in ISO 10116.		
FCS_COP.1/Hash	Controller, SE	The TOE supports hash functions using SHA-1, SHA-256, SHA-384, and SHA-512 within SSH. SHA-1, SHA-256 and SHA-384 are supported within TLS HMAC functions. SHA-512 is used for storing passwords in a non-plaintext form.		
FCS_COP.1/KeyedHash	Controller, SE	The TOE supports keyed hash message authentication using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512. HMAC-SHA-1 and HMAC-SHA-256 operate on 512-bit blocks, HMAC-SHA-384 and HMAC-SHA-512 operate on 1024-bit blocks of data, with key length of 256 bits, 512 bits and message digest sizes of 160-bits, 256-bits, 384-bits and 512 bits respectively.		
FCS_COP.1/SigGen	Controller, SE	The TOE supports signature generation and verification using RSA according to FIPS PUB 186-4 RSASSA-PKCS1v1_5 with at least 2048-bit and 3072-bit key sizes. The TOE also supports signature generation and verification using ECDSA according to FIPS PUB 186-4 with curves P-256, P-384, and P-521.		

FCS_HTTPS_EXT.1	Controller	The Controller component implements HTTPS for remote management in accordance with RFC 2818. The implementation conforms to all SHALL and SHOULD statements and its behavior does not conflict with any SHOULD NOT or SHALL NOT statements.
FCS_NTP_EXT.1	Controller	The Controller component implements NTP v4 for reliable time stamps in accordance with RFC 5905. Time updates are validated using authentication that uses SHA1 as the message digest algorithm. The Controller supports upto 5 NTP time sources. By default, The TOE rejects broadcast and multicast time updates.
FCS_RBG_EXT.1	Controller, SE	The TOE implements two instances of CTR_DRBG based on SP 800-90A (OpenSSL which are seeded with a min-entropy of 256 bits. The primary noise source(s) used for the Controller and the SE consists of CPU Jitter (Jent_v3.3.1).
FCS_SSHS_EXT.1	Controller	The TOE supports SSHv2 with public key and password-based authentication. When used in support of FTP_TRP.1, public key and password-based methods are supported. The supported authentication algorithms include ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp521. Same algorithms are also supported as host key algorithms. When acting as a server, the SSH client's presented public key must match one that is stored within the SSH TOE's authorized_keys file. Password-based authentication is supported by looking up the username in /etc/passwd and comparing the hash of the password to the value in /etc/shadow. If the credentials correspond to an entry in the files, the user is successfully authenticated and is authorized to access the TOE. The transport mechanisms support aes128-ctr and aes256-ctr for symmetric encryption and hmac-sha2-512 function for integrity. Key exchange is performed using ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512. Per RFC 4253, if a packet received is larger than 256 kilobytes, the connection will be rejected. When session keys are established, the TOE will monitor the length of time and number of packets transmitted using the same session key. Once the amount of time reaches one hour, or the total size of transmitted packets reaches one gigabyte, a rekey is initiated. There are no optional characteristics specified for FCS_SSH_EXT.1.4.

FCS_TLSC_EXT.1	Controller, SE	The TOE component (Controller) implements TLS v1.1 and v1.2 acting as a client. TLS/SSL versions 1.0 and earlier are rejected. The TOE supports the following cipher suites for connection between Controller and External IT entity (Audit server): • TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 • TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246 • TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246 • TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288 • TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 The cipher suites specified above are those listed in FCS_TLSC_EXT.1. The TOE component SE implements TLS v1.1 and v1.2 acting as a client. TLS/SSL versions 1.0 and earlier are rejected. The

		SE supports the following cipher suites for connection between SE and Controller entity (Connection between TOE components: • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 The reference identifier used for distributed TOE registration is a unique token generated by the Controller and requested by the administrator when deploying the SE. The reference identifier for intra-TOE communication is FQDN in CN. The reference identifier used for Controller syslog communications is based on DNS names per RFC 6125. The TOE supports the Supported Elliptic Curves extension with secp256r1, secp384r1, and secp521r1 and The TOE sends the public ECDHE component of its Key Exchange message during the TLS handshake. Out of these Supported Elliptic curves, SE supports only secp256r1. This is default configuration. The TOE supports the use of wildcard certificates, including for inter-TOE communication. The wildcard must be in the left-most label of the presented identifier. And the wildcard only covers one level of subdomains. For the reference identifier without a left-most label as in the certificate, the connection will fail, i.e., awesome.com doesn't match *.awesome.com.
FCS_TLSS_EXT.1	Controller	The TOE component (controller) implements TLS v1.2 and TLS v1.1 acting as a server. All other TLS/SSL versions are rejected. The TOE supports the following cipher suites for the Controller to Remote management system for WebUI: • TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 • TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

		• TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246 • TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246 • TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288 • TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 Controller to SE: • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 The TOE supports RSA with key size 2048 bits, 3072 bits and the ECDHE curves secp256r1, secp384r1 and secp521r1 of its Key Exchange message during the TLS handshake. By default, all parameters are preconfigured on the TOE when FIPS/CC mode is enabled. The TOE supports session resumption based on session tickets. Session tickets adhere to the structural format provided in section 4 of RFC 5077. The encryption algorithm specifically employed for encrypting session tickets is AES in

		Galois/Counter Mode (AES-GCM). When TOE is acting as a TLS Server, session tickets are protected with an AES-GCM algorithm with a key length of 256. In TLS session resumption, the server verifies the authenticity of presented session tickets. If validation is unsuccessful, it prompts the server to initiate a full handshake process. A distinct context is established for every session, ensuring there is no interaction between them even if they are created for the same user.
FIA_AFL.1	Controller	The Controller administrative Web UI and remote SSH CLI are capable of detecting and tracking login failures related to Administrators attempting to authenticate remotely using a password. Once a configurable threshold between 3 and 20 attempts is reached, the account will be locked until a configurable time period (default of 300 seconds) has elapsed. The Controller local console is not subject to authentication failure lockout mechanism, which ensures that remote authentication failures (via SSH or HTTPS) cannot lead to a situation where no administrator access is available.
FIA_PMG_EXT.1	Controller	The Controller is the administrative component of the TOE which maintains administrator passwords. Passwords must be of length configurable between 8 and 32 characters and may be composed of uppercase and lowercase letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”,”-“, and “+”.
FIA_UIA_EXT.1 FIA_UAU_EXT.2 FIA_UAU.7	Controller	For remote administration, a login form is presented on the Controller HTTPS Web UI, where the administrator must present a valid username and password before access is permitted. If either value is not correct, the authentication fails, and the counter is incremented. If the values are correct, the administrator is redirected to the Console Web UI. For SSH, the user may supply a username and password or username and public key for authentication. If either the username, password or public keys are not valid for the administrator attempting authentication, the attempt fails, and the login prompt is shown again. If successful, the CLI command prompt is shown. For local authentication to the console, the username and password prompts are given, and if they do not match, the console returns to the login prompt. If the username and password combination is correct, the CLI command prompt is shown. In all cases, while no actions are possible before authentication, the configured access banner is presented to GUI, SSH, Local console users before identification.

		The TOE detects unsuccessful authentication attempts related to Administrators attempting to authenticate remotely using a password and will increment the counter. Failed authentication using public key via SSH will not increment the counter. The password must contain at least one occurrence of three of the following four categories: upper-case letters, lower-case letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, ["_", "+" and the length should be configured between 8 to 32 characters. During authentication to the local and remote Controller cli, password characters are obfuscated by not providing any visual feedback of the characters being typed. For the remote GUI, passwords are obfuscated by using a dot-like (●) character. Password obfuscation for all remote and local methods is default with no configuration required. The Service Engine does not have any authentication and identification interface. Only the Controller has an authentication and identification interface. All the configuration of the SE is done via the Controller’s interface.
FIA_X509_EXT.1/Rev	Controller	The TOE supports X.509 certificates for authentication of trusted channel which is syslog servers. During the TLS connection establishment, the TOE will check the certificate chain presented by the remote peer to ensure that: 1) All certificates in the chain are not expired 2) All certificates terminate with a root CA which is installed in the TOE’s trust store 3) All certificates contain the proper key usage arguments. More specifically: a. Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field. b. Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field. 4) All CA certificates contain the basicConstraints flag which is set to TRUE The ability to modify or delete trust store certificates is restricted to authorized administrators only. Certificate revocation is conducted for both leaf and intermediate certificates. On the controller, customers can furnish the CRL through either a file or by specifying a server

		URL containing the CRL. In a later scenario, a job manager operates at the customer-defined interval to update the CRL with minimum time being 30 mins. Revocation checks are applied to both leaf and intermediate certificates. Regardless of whether the client provides only the leaf certificate or the entire certificate chain, the server employs the **'ssl_client_certificate'** directive to designate trusted CA's or intermediate certificates. During this process, the server verifies the revocation status of each certificate provided.
FIA_X509_EXT.1/ITT	SE	The SE component validates the Controller server's X.509 certificate during internal distributed TOE communications. During the TLS handshake, the SE component will check the certificate chain presented by the Controller to ensure that: • All certificates in the chain are not expired • All certificates terminate with a root CA which is installed in the SE trust store • All certificates contain the proper key usage arguments. More specifically: • Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field. • All CA certificates contain the basicConstraints flag which is set to TRUE Revocation checking is not performed on the internal TLS connection between the Controller and the SE.
FIA_X509_EXT.2	Controller	Each TOE component maintains its own internal trust store which may only contain a single end-entity cert, which will be used for any operations requiring certificates. Certificates are used by the controller for HTTPS/TLS server authentication to remote management workstations and to SEs (TOE component) for inter TOE communication channel.CA certificates along with a configured reference identifier is used by the controller to authenticate an external audit server. If the revocation server cannot be reached during the validity check for any connection (with the exception of FPT_ITT.1), the certificate will not be accepted, and the connection will fail.
FIA_X509_EXT.3	Controller	The TOE generates Certificate Request as specified by RFC 2986 and provides the public key and Common Name, Organization, Organizational Unit, and Country in the request.

		The TOE validates the chain of certificates from the Root CA upon receiving the CA Certificate Response.
FMT_MOF.1/Functions FMT_MOF.1/ManualUpdate FMT_MOF.1/Services FMT_MTD.1/CoreData FMT_MTD.1/CryptoKeys FMT_SMF.1 FMT_SMR.2	Controller, SE	All configuration of security management functionality is restricted to an authorized administrator via the Controller component, which includes following functions: • Ability to administer the TOE locally and remotely; • Ability to configure the access banner; • Ability to configure the sessionactivity time before session termination or locking; • Ability to update the TOE, and to verify the updates using [hash comparison] capability prior to installing those updates; • Ability to configure the authentication failure parameters for FIA_AFL.1; • Ability to start and stop services; • Ability to modify the behaviour of the transmission of audit data to an external IT entity; • Ability to manage the cryptographic keys; • Ability to configure the cryptographic functionality; • Ability to configure the interaction between TOE components; • Ability to set the time which is used for time-stamps; • Ability to configure NTP; • Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors; • Ability to import X.509v3 certificates to the TOE's trust store; • Ability to manage the trusted public keys database; No administrative actions are available prior to administrator authentication. Using the Controller, the administrator can generate and specify the key pairs used for requesting TLS certificates. Cryptographic keys for the SE are generated by the administrator, using the Controller. Only administrators can manage the trust store. The ability to modify, add, import, export, or delete trust store certificates, and designating certificates as trust anchors is restricted to authorized administrators only. All management functions as specified in FMT_SMF.1 for the TOE are carried out via the Controller Web GUI (remote) and CLI (remote and local). The SE component polls the

		Controller component every 30 seconds and applies any new configuration changes (if any were applied since the last update). There are two types of users, admin and non-admin, with different privileges as per the AGD. The users have access based on the Tenant and Role selected and can perform task as per its role. The TOE includes a Security Administrator role which configures, monitors, and administers the TOE. All TSF-related functions are managed using this role. The local administrative interface for the TOE (Controller) is the interactive CLI console available through ESXi.
FPT_APW_EXT.1	Controller	Administrative passwords are stored in the Controller filesystem as a salted hash value with SHA-512 so that it cannot be viewed.
FPT_ITT.1 FPT_ITT.1/Join	Controller, SE	The Controller and the SE communicate over TLS. In this case, Controller is the TLS server, and the SE is the TLS client. The controller ID is tied to a Security token to a particular SE. The controller generates the Token and passes the Token to the SE as an env variable. After the SE comes up, it communicates the controller through TLS and the token is passed post the TLS connection. SE knows that it is talking to the controller based on the OVA and server certificates. Tokens and certificates are used for joining components. SE always initiates the connection. If token verification fails, TOE rejects the SE (component). The Security administrator must build OVA file with new security token to deploy SE. The Security Administrator uses the Controller component to register the SE endpoints user and its private and public key. The Security Administrator configures the SE user and its corresponding public key on the SE according to instructions in the VMware Avi Load Balancer Administration Manual for Common Criteria. This enables the SE to establish communication with the Controller and to receive configuration updates.
FPT_SKP_EXT.1	Controller, SE	All the locally stored keys are encrypted using AES256-CBC, except the keys used for inter-TOE communication. Controller- SE communication keys are unencrypted and stored in the underlying filesystem. The TOE does not provide any functionality to display or export any of the keys through any TOE interface.
FPT_STM_EXT.1	Controller, SE	The time can be manually configured by the Security Administrator via CLI (both SSH and console) and NTP is also supported by the Controller and can be configured by WebUI. The SE depends on the controller for time source.

		The system clock is used for the following functions: • Timestamps for audit events • X.509 certificate validity check • Session timeout duration • Authentication failure lockout duration • SSH and TLS session timeout durations
FPT_TST_EXT.1	Controller, SE	The TSF runs a suite of the following self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF: • Software integrity test; • OpenSSL crypto module Known Answer Tests (KATs). Both the TOE components perform these self-tests during initial start-up (on power on). For testing of the TSF, the TOE automatically runs checks and tests at startup, to ensure the TOE is operating correctly, including checks of image integrity and the cryptographic functionality. Software Integrity Test – The Software Integrity Test runs automatically whenever the system image is loaded and confirms that the image file that's about to be loaded has maintained its integrity using HMAC SHA-1. Software Integrity Check compares the runtime calculated value against the pre-computed known value using HMAC SHA-1. If any component reports failure for the POST, the system crashes and appropriate information is displayed on the screen and the system stops/halts. The TOE administrator must report to VMware TAC team. For the OpenSSL KATs, the cryptographic implementation of the TOE is tested to verify that for each cryptographic algorithm, it produces the expected results against the expected output. This includes KATs for all cryptographic algorithms implemented by the TSF: ○ AES encryption / decryption test (KAT) ○ Hash KAT ○ HMAC KAT ○ RSA signature test (Pairwise Consistency Test) ○ ECDSA signature test (Pairwise Consistency Test) ○ DRBG health check (KAT)

		A KAT (Known Answer Test) test is a test where a cryptographic algorithm is run on data for which the correct output is already known. The calculated output is compared with the known answer to determine the correctness of the implementation. A PCT (Pairwise Consistency Test) test is run when an asymmetrical key pair is generated. It uses the public key to encrypt a plaintext, and uses the private key to decrypt the encrypted text. If the decryption is successful, the test succeeds. Otherwise, the test fails. During the system bootup process (power on or reboot), all the Power on Startup Test (POST) components for the cryptographic modules perform the POST for the corresponding component (hardware or software).Ensuring that all components of the TOE software and its cryptographic implementation have not been altered and are working as expected is sufficient to demonstrate that the TSF is operating correctly.
FPT_TUD_EXT.1	Controller, SE	The TOE supports manual updates for each TOE component. The active version of the TOE is queried from the Controller Web GUI Monitoring screen for each component. The TOE maintains the active version and a system default version. Each TOE component is shipped with a manufacturing release which is a system default build. When the TOE is upgraded to a new version, it will overwrite the previous version (if different from the system default build). The new version becomes the active version once the installation process is completed. The TOE does not support installation of trusted updates with delayed activation. When the TOE is reset to factory defaults, any newer versions are deleted, and the TOE software is replaced with the system default build. The manual update process is initiated through the Controller for all TOE components. Published hash (MD5 and SHA-256) mechanisms are used to verify software/firmware update files (to ensure they have not been modified from the originals distributed by VMware) before they are used to actually update the applicable TOE components. The verification is not performed on the TOE during the update process. The Security Administrator manually verifies the published hash prior to initiating the update process. If hash verification fails the, software/firmware is not to be used for an update.

		After successful verification of public hash, the image is ready to be uploaded on the TOE. There are no functions which may cease to operate during a manual update. The update is initiated on the Controller, which is updated first, followed by Service Engines. Candidate updates are obtained through the https://my.vmware.com user portal. Only a user with the appropriate entitlement can download the software.
FTA_SSL.3	Controller	The Controller Web UI and SSH CLI sessions are terminated after an administrator configurable time period. The default value is 15 minutes.
FTA_SSL.4	Controller	Remote administrative sessions are terminated using the 'sign out' link from the Controller Web UI, while both remote and local CLI sessions are terminated using the "exit" command.
FTA_SSL_EXT.1	Controller	Local console session timeout depends on the vCenter Administration. So, the vCenter admin has to configure the settings. After logging into the controller via SSH, Security administrator has to enter into the Shell mode (Shell mode is the TOE's proprietary mode). Once entered into the 'shell' mode, security administrator configures the timeout, and the shell session gets terminated after expiring the administrator configurable time period.
FTA_TAB.1	Controller	The Controller Web UI (HTTPS), instead of CLI (SSH and local console) is configured by an administrator to display an advisory notice and consent warning message prior to identification and authentication. Once configured, the same message appears over all interfaces (Web UI and CLI).
FTP_ITC.1	Controller, SE	The TOE communicates with the external syslog server using Syslog over TLS with Authentication as described in the descriptions of FAU_STG_EXT.1. The TSF initiates the trusted channel with the Syslog server, ensuring assured identification of the Syslog server based on the X.509 certificates. The Controller component acts as a client for communications with IT entities: • TLS v1.1/v1.2 (Syslog server)
FTP_TRP.1/Admin	Controller	The Controller component supports trusted channels secured using SSHv2 and HTTPS/TLSv1.1/1.2 for remote management.

6.1 CAVP Algorithm Certificate Details

Each of these cryptographic algorithms have been validated as identified in the table below.

Table 13 - CAVP certificates

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	RSA KeyGen (Modulo 2048, 3072)	A1292
	ECC schemes using “NIST curves” [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	ECDSA KeyGen (Curve P-256,P-384,P-521) ECDSA KeyVer (Curve P-256,P-384,P-521)	A1292
	FFC Schemes using ‘safe-prime’ groups that meet the following: “NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [RFC 3526,]	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	Safe Primes Key Generation	Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1. No CAVP Certificate.
FCS_CKM.2	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1”	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	Vendor Affirmed	No NIST CAVP, CCTL has performed all assurance/evaluation activities and documented in the ETR, DTRs, and

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
				AAR accordingly.
	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	KAS-ECC-SSC Sp800-56Ar3 (Domain Parameter Generation Methods: P-256, P-384, P-521)	A1292
	FFC Schemes using “safe-prime” groups that meet the following: ‘NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [groups listed in RFC 3526,]	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	KAS-FFC-SSC Sp800-56Ar3 (Domain Parameter Generation Methods: FB, FC)	A1292
FCS_COP.1/ DataEncryption	AES used in [CBC, CTR, GCM] mode and cryptographic key sizes [128 bits, 256 bits]	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	AES-CBC (Key length 128,256) AES-CTR (Key length 128,256) AES-GCM (Key length 128,256)	A1292
FCS_COP.1/ SigGen	For RSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	RSA SigGen (Modulo: 2048, 3072) RSA SigVer (Modulo: 2048, 3072)	A1292
	For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [P-	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	ECDSA SigGen (Curve: P-256, P-384, P-521)	A1292

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #
	256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4		ECDSA SigVer (Curve: P-256, P-384, P-521)	
FCS_COP.1/ Hash	[SHA-1, SHA-256, SHA-384, SHA-512] and message digest sizes [160, 256, 384, 512] bits	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	SHA-1, SHA-256, SHA-384, SHA-512	A1292
FCS_COP.1/ KeyedHash	[HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512,] and cryptographic key sizes [256 bits, 512 bits] and message digest sizes [160, 256, 384, 512] bits	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	A1292
FCS_RBG_EXT.1	CTR_DRBG (AES)	VMware's OpenSSL FIPS Object Module, version 2.0.20-vmw	DRBG (Mode: AES-256)	A1292

6.2 Distributed TOE SFR Allocation

For a distributed TOE, the SFR in the PP as well as any relevant EPs/Modules/Packages must be met by the TOE as a whole. However, each TOE component will not necessarily meet each SFR. specifies when each SFR must be implemented by a component.

The following categories are used to define those SFR allocations:

- All Components (All): All components that comprise of the distributed TOE must independently satisfy the requirement.
- At least one Component (One): This requirement must be fulfilled by at least one component within the distributed TOE.
- Feature Dependent (Feature Dependent): These requirements will only be fulfilled where the feature is implemented by the distributed TOE component.

Table 14 – Distributed TOE SFR Allocation

Requirement	SFR Allocation	Controller	SE
FAU_GEN.1	All	X	X
FAU_GEN.2	All	X	X
FAU_GEN_EXT.1	All	X	X
FAU_STG.1	Feature Dependent	X	
FAU_STG_EXT.1	All	X	X

Requirement	SFR Allocation	Controller	SE
FAU_STG_EXT.4	Feature Dependent	X	
FAU_STG_EXT.5	Feature Dependent		X
FCO_CPC_EXT.1	All	X	X
FCS_CKM.1	One	X	X
FCS_CKM.2	All	X	X
FCS_CKM.4	All	X	X
FCS_COP.1/DataEncryption	All	X	X
FCS_COP.1/SigGen	All	X	X
FCS_COP.1/Hash	All	X	X
FCS_COP.1/KeyedHash	All	X	X
FCS_HTTPS_EXT.1	Feature Dependent	X	
FCS_NTP_EXT.1	Feature Dependent	X	
FCS_SSHS_EXT.1	Feature Dependent	X	
FCS_TLSC_EXT.1	Feature Dependent	X	X
FCS_TLSS_EXT.1	Feature Dependent	X	
FCS_RBG_EXT.1	All	X	X
FIA_AFL.1	One	X	
FIA_PMG_EXT.1	One	X	
FIA_UIA_EXT.1	One	X	
FIA_UAU_EXT.2	One	X	
FIA_UAU.7	Feature Dependent	X	
FIA_X509_EXT.1/Rev	Feature Dependent	X	
FIA_X509_EXT.1/ITT	Feature Dependent		X
FIA_X509_EXT.2	Feature Dependent	X	
FIA_X509_EXT.3	Feature Dependent	X	
FMT_MOF.1/Functions	Feature Dependent	X	
FMT_MOF.1/ManualUpdate	All	X	X
FMT_MOF.1/Services	All	X	X
FMT_MTD.1/CoreData	All	X	X
FMT_MTD.1/CryptoKeys	Feature Dependent	X	
FMT_SMF.1	Feature Dependent	X	
FMT_SMR.2	All	X	X
FPT_SKP_EXT.1	All	X	X
FPT_APW_EXT.1	Feature Dependent	X	
FPT_TST_EXT.1	All	X	X

Requirement	SFR Allocation	Controller	SE
FPT_ITT.1 FPT_ITT.1/Join	Feature Dependent	X	X
FPT_STM_EXT.1	All	X	X
FPT_TUD_EXT.1	All	X	X
FTA_SSL_EXT.1	Feature Dependent	X	
FTA_SSL.3	Feature Dependent	X	
FTA_SSL.4	Feature Dependent	X	
FTA_TAB.1	One	X	
FTP_ITC.1	One	X	
FTP_TRP.1/Admin	One	X	

6.3 Cryptographic Key Destruction

The table below describes the key zeroization provided by the TOE and as referenced in FCS_CKM.4.

Table 15 – Cryptographic Key Destruction

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
TLS server private key	This key is used for authentication, so the server can prove who it is. The private key is used for TLS secure connections.	HDD and Postgress SQL DB	It is present till the life of configuration of Server. Automatically when the server configuration is changed on the TOE.
TLS server public key	This key is used to encrypt the data that is used to compute the secret key. The public key is used for TLS secure connection.	HDD and Postgress SQL DB	It is present till the life of configuration of Server. Automatically when the server configuration is changed on the TOE.
TLS pre-master secret	The pre-master secret is the client and server exchange of random numbers and a special number, the pre-master secret. This pre-master secret uses asymmetric cryptography from which new TLS session keys can be created.	HDD and Postgress SQL DB	It is present till the life of configuration of Server. Automatically when the server configuration is changed on the TOE.

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
TLS session encryption key	The session encryption key is unique for each session and is based on the shared secrets that were negotiated at the start of the session. The Key is used to encrypt TLS session data.	RAM	Till the lifetime of session. Automatically after the session complete
TLS session integrity key	This key is used to provide privacy and TLS data integrity protection.	RAM	Till the lifetime of session. Automatically after the session complete
SSH Private Key	This key is used for Server authentication	HDD	Autogenerated when the controller comes up. Automatically when the server configuration is changed on the TOE.
SSH Session Key	The session encryption key is unique for each session and is based on the shared secrets that were negotiated at the start of the session. The Key is used to encrypt SSH session data.	RAM	Till the lifetime of session of sshd in TOE.
Diffie-Hellman Shared Secret	This is the shared secret used as part of the Diffie-Hellman key exchange.	RAM	Till the lifetime of session. Automatically after the session complete.
Diffie Hellman private exponent	This is the private exponent used as part of the Diffie-Hellman key exchange.	RAM	Till the lifetime of session. Automatically after the session complete.
NTP Keys	This is the key that is used for the encryption and decryption of authentication of NTP packets.	RAM	Autogenerated when the controller comes up. Persists, till the lifetime of controller and the configuration is unchanged.

7 Acronym Table

Acronyms should be included as an Appendix in each document.

Table 16 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CC	Common Criteria
CRL	Certificate Revocation List
DTLS	Datagram Transport Layer Security
EP	Extended Package
GUI	Graphical User Interface
IP	Internet Protocol
NDcPP	Network Device Collaborative Protection Profile
NIAP	Nation Information Assurance Partnership
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OVA	Open Virtual Appliance
PP	Protection Profile
RSA	Rivest, Shamir & Adleman
SE	Service Engine
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSS	TOE Summary Specification