Ivanti Connect Secure 22.2 Security Target

Document Version: 0.6

intertek
acumen
security

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History

Version	Date	Changes
Version 0.1	May 27, 2022	Initial Release
Version 0.2	June 9, 2022	Updated requirements based on responses from vendor
Version 0.3	November 14, 2022	Updated requirements based on Testing
Version 0.4	April 06. 2023	Updated based on check-in ECRs
Version 0.5	October 10, 2023	Minor update
Version 0.6	December 12, 2023	Minor update

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	Ivanti Connect Secure 22.2 Security Target
ST Version	0.6
ST Date	December 12, 2023
ST Author	Intertek Acumen Security
TOE Hardware	ISA Models 6000, 8000C, 8000F and Virtual Appliance
TOE Identifier	Ivanti Connect Secure 22.2
TOE Version	22.2R3
TOE Developer	Ivanti, Inc. 10377 South Jordan Gateway, Suite 110 South Jordan, Utah 84095
Key Words	Network Device, Network Virtual Appliance

1.2 Product Overview

1.2.1 Product Type

Ivanti Connect Secure enables secure device access to applications and resources in the data center and in the cloud. This clientless system provides unified management of policies, compliance, and authorizations for cloud and data center access. The single sign-on and Layer 3 SSL VPN functionality of the Product are outside the scope of this evaluation. For a list of product features and functionality that is excluded from the evaluation, please refer to Section **Error! Reference source not found.**.

1.2.2 TOE Usage

The TOE is classified as a network device (a generic infrastructure device that can be connected to a network) or a virtual network device (a Virtual Appliance that can be connected to a network) depending on the underlying platform. The TOE software consists of Ivanti Connect Secure (ICS) 22.2R3. The appliance's software is built on IVE OS 3.0. The TOE consists of the ICS application, IVE OS, and either the TOE hardware or the VM hypervisor, all of which are delivered with the TOE. The TOE hardware consists of either the ISA Models 6000, 8000C, or 8000F.

The TOE provides following security features that are part of the evaluated configuration:

- Secure remote administration of the TOE via HTTPS/TLS web interface

- Secure Local administration of the TOE
- Secure connectivity with remote audit servers using mutually authenticated TLS
- Identification and authentication of the administrator of the TOE
- CAVP validated cryptographic algorithms
- Self-protection mechanisms such as executing self-tests to verify correct operation
- Secure firmware updates

For a complete list of security features provided by the TOE, please refer to Section **Error! Reference source not found.**.

1.3 TOE Description

This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references. In the below diagram, the TOE consists of the appliance within the blue line. Everything else is not included within the TOE and is part of the TOE environment.

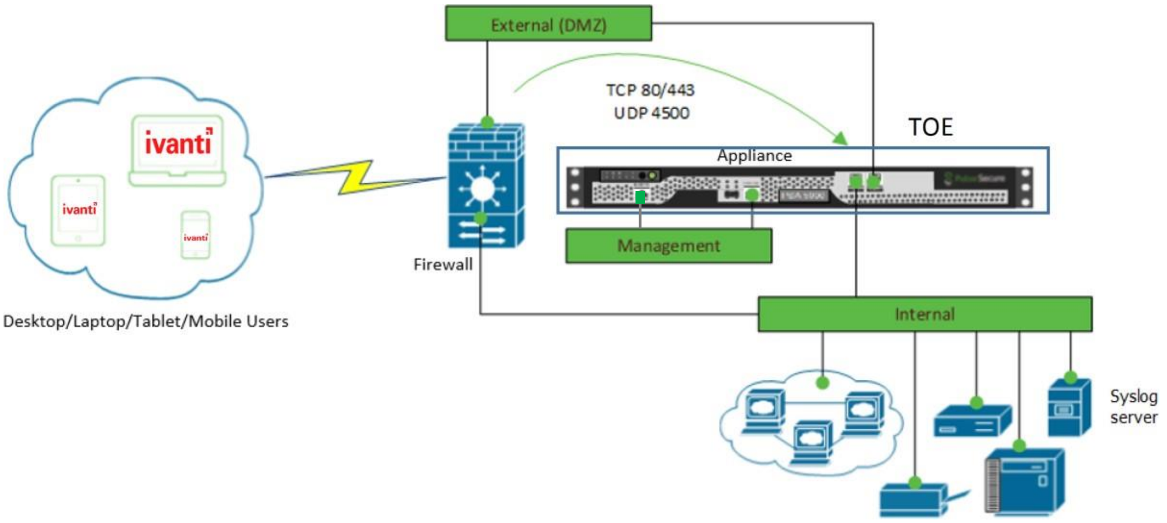

Figure 1 – Representative TOE Deployment

1.3.1 Physical Boundaries

The TOE consists of the following hardware:

- ISA 6000
- ISA 8000C
- ISA 8000F

Running:

- Ivanti Connect Secure (ICS) v22.2R3

The TOE can also be a virtual appliance (ISA-V) on VMware ESXi 6.7, with a Dell PowerEdge R640 as the hardware platform. The ICS software runs on any of the TOE hardware appliance platforms or on a virtual appliance. The TOE is delivered with the ICS v22.2R3 software installed on one of the ISA appliances. The platforms provide different amounts of processing power and network connectivity options as described in Table 2.

Table 2 – TOE Hardware Details

Model	Processor	Network Options
ISA 6000	Intel Core i3 10100E 10th gen (Comet Lake)	2 x 10 Gigabit Copper Ethernet traffic ports 1 x 1GbE Management port 1 x RJ-45 Console Port
ISA 8000C	Intel Xeon Gold 5317 (Ice Lake)	2 x 10 Gigabit Ethernet copper traffic ports with link redundancy 1 x 1GbE Management port 1 x RJ-45 Console Port
ISA 8000F	Intel Xeon Gold 5317 (Ice Lake)	2 x 10 Gigabit fiber trafficports with link redundancy 1 x 1GbE Management port 1 x RJ-45 Console Port

The TOE can also be a virtual appliance on VMware ESXi 6.7, with a Dell PowerEdge R640 as the hardware platform. ESXi is a bare-metal hypervisor so there is no underlying operation system. In the evaluated configuration, there are no guest VMs on the physical platform providing non-network device functionality. The virtual appliance platform is described below.

The virtual appliance can be download by customers from https://my.pulsesecure.net/ and installed on compliant hardware listed below. License are provided by Ivanti Secure via email. When a customer request is received, Ivanti will provide an authcode via email. Customers must register in https://my.pulsesecure.net/ portal and generate the license string by providing Hardware id with earlier provided authcode. These auth codes are not reusable.

Table 3 - VMware Host Details

Model	Processor	Hypervisor
ISA-V (virtual platform) on PowerEdge R640	Intel(R) Xeon(R) Gold 6252 CPU @ 2.10GHz	VMware ESXi 6.7

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 TOE generates audit records for security relevant events. The TOE maintains a local audit log as well as sending the audit records to a remote Syslog server. Audit records sent to the remote server are protected by a TLS connection. Each audit record includes identity (username, IP address, or process), date and time of the event, type of event, and the outcome of the event. The TOE prevents modification to the local audit log.

1.3.2.2 Cryptographic Support

The TOE includes the Ivanti Secure Cryptographic Module that implements CAVP-validated cryptographic algorithms for random bit generation, encryption/decryption, authentication, and integrity protection/verification. These algorithms are used to provide security for the TLS and HTTPs connections for secure management and secure connections to a syslog server. TLS and HTTPs are also used to verify firmware updates. The cryptographic services provided by the TOE are described below.

Table 4 TOE Cryptographic Protocols

Cryptographic Protocol	Use within the TOE
TLS (client)	Secure connection to syslog FCS_TLSC_EXT.1, FCS_TLSC_EXT.2
HTTPS/TLS (server)	Secure management connections and verification of firmware updates via web browser FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1
AES	Provides encryption/decryption in support of the TLS protocol. FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1
DRBG	Deterministic random bit generation used to generate keys. FCS_TLSS_EXT.1, FCS_RBG_EXT.1
Secure hash	Used as part of digital signatures and for hashing passwords prior to storage on the TOE. FCS_COP.1/Hash, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FPT_APW_EXT.1
HMAC	Provides keyed hashing services in support of TLS. FCS_COP.1/KeyedHash, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1
ECDSA	Provides key generation and signature generation and verification in support of TLS. FCS_CKM.1, FCS_COP.1/SigGen, FCS_COP.1/SigVer, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1
EC-DH	Provides key establishment for TLS. FCS_CKM.2, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1
RSA	Provide key generation and signature generation and verification (PKCS1_V1.5) in support of TLS. FCS_CKM.1, FCS_COP.1/SigGen, FCS_COP.1/SigVer, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1

Each of these cryptographic algorithms have been validated for conformance to the requirements specified in their respective standards, as identified in Section 6.1 CAVP Algorithm Certificate Details.

1.3.2.3 Identification and Authentication

The TOE authenticates administrative users using a username/password or username/X.509 certificate combination. The TOE does not allow access to any administrative functions prior to successful authentication. The TOE validates and authenticates X.509 certificates for all certificate uses.

The TOE supports passwords consisting of alphanumeric and special characters and enforces minimum password lengths. The TSF supports certificates using RSA or ECDSA signature algorithms.
The TOE only allows users to view the login warning banner and send/receive ICMP packets prior to authentication.

Remote administrators are locked out after a configurable number of unsuccessful authentication attempts.

1.3.2.4 Security Management

The TOE allows users with the Security Administrator role to administer the TOE over a remote web UI or a local CLI. These interfaces do not allow the Security Administrator to execute arbitrary commands or executables on the TOE. Security Administrators can manage connections to an external Syslog server, as well as determine the size of local audit storage.

1.3.2.5 Protection of the TSF

The TOE implements several self-protection mechanisms. It does not provide an interface for the reading of secret or private keys. The TOE ensures timestamps, timeouts, and certificate checks are accurate by maintaining a real-time clock. Upon startup, the TOE runs a suite of self-tests to verify that it is operating correctly. The TOE also verifies the integrity and authenticity of firmware updates by verifying a digital signature of the update prior to installing it.

1.3.2.6 TOE Access

The TOE can be configured to display a warning and consent banner when an administrator attempts to establish an interactive session over the local CLI or remote web UI. The TOE also enforces a configurable inactivity timeout for remote and local administrative sessions.

1.3.2.7 Trusted Path/Channels

The TOE uses TLS to provide a trusted communication channel between itself and remote Syslog servers. The trusted channels utilize X.509 certificates to perform mutual authentication. The TOE initiates the TLS trusted channel with the remote server.

The TOE uses HTTPS/TLS to provide a trusted path between itself and remote administrative users. The TOE does not implement any additional methods of remote administration. The remote administrative users are responsible for initiating the trusted path when they wish to communicate with the TOE.

1.3.3 TOE Documentation

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

- Ivanti Secure Operational User Guidance and Preparative Procedures
- Ivanti Connect Secure Administration Guide: https://help.ivanti.com/ps/help/en_US/ICS/22.x/ag/landingpage.htm
- Ivanti Connect Secure Supported Platforms Guide: https://help.ivanti.com/ps/help/en_US/ICS/22.x/spg/Default.htm
- Ivanti Connect Secure Security Target

1.3.4 References

In additional to TOE documentation, the following reference may also be valuable when understanding and controlling the TOE:

- Collaborative Protection Profile for Network Devices Version 2.2e

1.4 TOE Environment

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

Table 5 – Required Environmental Components

Components	Required	Description
Syslog server	Yes	• Conformant with RFC 5424 (Syslog Protocol) • Supporting Syslog over TLS (RFC 5425) • Acting as a TLSv1.1 and/or TLSv1.2 server • Supporting Client Certificate authentication • Supporting at least one of the following cipher suites: ○ TLS_RSA_WITH_AES_128_CBC_SHA ○ TLS_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA ○ TLS_RSA_WITH_AES_128_CBC_SHA256 ○ TLS_RSA_WITH_AES_256_CBC_ SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

Components	Required	Description
Management laptop with web browser	Yes	• Provides remoted management of TOE • Microsoft Edge 101, Google Chrome 102, or Firefox 100 • Supporting TLSv1.1 and/or TLSv1.2 • Supporting at least one of the following ciphersuites: ○ TLS_RSA_WITH_AES_128_CBC_SHA ○ TLS_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA ○ TLS_RSA_WITH_AES_128_CBC_SHA256 ○ TLS_RSA_WITH_AES_256_CBC_ SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
CRL Server	Yes	• Conformant with RFC 5280
DNS Server	Yes	• Conformant with RFC 1035

1.5 Product Functionality not Included in the Scope of the Evaluation

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

- Layer 3 SSL VPN
- Application VPN
- Endpoint Integrity and Assessment
- Layer 7 Web single sign-on (SSO) via SAML
- Mobile Device Management Integration

These features may be used in the evaluated configuration; however, no assurance as to the correct operation of these features is provided.

The TOE includes the following functionality that is not covered in this Security Target and may not be enabled or used in in the CC evaluated configuration:

- DMI Agent
- SNMP Traps
- External Authentication Servers for administrator authentication

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, 27 March 2020 [PP-ND]

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 Protection Profile (PP) , 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 6 identifies all applicable TDs.

Table 6 – 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	No	This TD addresses NTP functionality and this TOE does not support NTP.
TD0536: NIT Technical Decision for Update Verification Inconsistency	Yes	
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes	
TD0546: NIT Technical Decision for DTLS – clarification of Application Note 63	No	This TD addresses DTLS functionality and this TOE does not support DTLS functionality.
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
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	Yes	
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	
TD0592: NIT Technical Decision for Local Storage of Audit Records	Yes	
TD0631 – NIT Technical Decision for Clarification of public key authentication for SSH Server	No	This TD addresses SSH functionality, and this TOE does not include SSH functionality.
TD0632 – NIT Technical Decision for Consistency with Time Data for vNDs	Yes	
TD0633 – NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	No	This TD addresses IPSec functionality, and this TOE does not include IPSec functionality.

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
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	This TD addresses SSH functionality, and this TOE does not include SSH functionality.
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 for Link to Allowed-With List	Yes	
TD0792 – NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Yes	

3 Security Problem Definition

The security problem definition has been taken directly from the claimed PP 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 7 are drawn directly from the claimed PP.

Table 7 – 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 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.

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

3.2 Assumptions

The assumptions included in Table 8 are drawn directly from PP and any relevant EPs/Modules/Packages.

Table 8 – 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.
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).

ID	Assumption
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.
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	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.

ID	Assumption
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 9 are drawn directly from the claimed PP.

Table 9 – 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 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 below, track with the assumptions about the TOE operational environment.

Table 10 – 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_ADMN	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.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 11 – SFRs

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG.1	Protected Audit Trail Storage
FAU_STG_EXT.1	Protected Audit Event Storage
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_RBG_EXT.1	Random Bit Generation
FCS_TLSC_EXT.1	TLS Client Protocol without Mutual Authentication
FCS_TLSC_EXT.2	TLS Client Support for 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.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1/Functions	Management of Security Functions Behaviour
FMT_MOF.1/ManualUpdate	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

Requirement	Description
FPT_TST_EXT.1	TSF Testing
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;
- 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) 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 12.

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 12.

Table 12 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG.1	None	None
FAU_STG_EXT.1	None	None
FCS_CKM.1	None	None
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_RBG_EXT.1	None	None
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	None	None
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.2	None	None

Requirement	Auditable Events	Additional Audit Record Contents
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_MTD.1/CoreData	None	None
FMT_MTD.1/CryptoKeys	None	None
FMT_SMF.1	All management activities ofTSF data	None
FMT_SMR.2	None	None
FPT_SKP_EXT.1	None	None
FPT_APW_EXT.1	None	None
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
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

Requirement	Auditable Events	Additional Audit Record Contents
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_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.

5.2.1.4 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 consist of a single standalone component that stores audit data locally].

FAU_STG_EXT.1.3
The TSF shall [overwrite previous audit records according to the following rule: *[the oldest log file is overwritten by the new log file]*] when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 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] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4*;*

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

5.2.2.2 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";

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

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

5.2.2.3 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 [*[3]-* pass] overwrite consisting of [a pseudorandom pattern using the TSF's RBG];

  ]

that meets the following: *No Standard*

5.2.2.4 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, 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, GCM as specified in ISO 19772].

5.2.2.5 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, 3072 bits*]
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits, 384 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]; ISO/IEC 14888-3, Section 6.4

].

5.2.2.6 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*.

5.2.2.7 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] and cryptographic key sizes *[160 bits, 256 bits, 384 bits used in HMAC]* **and message digest sizes [160, 256, 384] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”*.

5.2.2.8 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 the HTTPS protocol using TLS.

FCS_HTTPS_EXT.1.3
If a peer certificate is presented, the TSF shall [not establish the connection] if the peer certificate is deemed invalid.

5.2.2.9 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 [*[3]* 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.

5.2.2.10 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_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

] 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, IPv4 address in CN or SAN*] 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 overridemechanism].

FCS_TLSC_EXT.1.4

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

5.2.2.11 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

FCS_TLSC_EXT.2.1

The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

5.2.2.12 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_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

] 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] and no other curves]].

FCS_TLSS_EXT.1.4
The TSF shall support [no session resumption or session tickets].

5.2.3 Identification and Authentication (FIA)

5.2.3.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within *[3 to 10]*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*]*.

5.2.3.2 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: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, *[and standard printable ASCII characters(values 0x20 – 0x7E)]*];

b) Minimum password length shall be configurable to between [*15*] and [*15*] characters.

5.2.3.3 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;

- [*Respond to ICMP Echo messages with an ICMP Echo Reply message*].

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.

5.2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

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

5.2.3.5 FIA_UAU.7.1 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**.

5.2.3.6 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 updatesand 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 extendedKeyUsagefield.*
  - *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.

5.2.3.7 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 [accept the certificate].

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

5.2.3.8 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.

5.2.4 Security Management (FMT)

5.2.4.1 FMT_MOF.1/Functions Management of Security Functions Behaviour

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

5.2.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behavior

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

5.2.4.3 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.*

5.2.4.4 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*.

5.2.4.5 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 [digital signature, hash comparison] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*

- [
    - Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);
    - 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 set the time which is used for time-stamps;
    - 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;
      ].

5.2.4.6 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.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 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.

5.2.5.2 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.

5.2.5.3 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].

5.2.5.4 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: [

- *BIOS checks*
  - *Verify boot block checksum*
  - *Verify main BIOS checksum*
  - *Check COMS diagnostic byte to determine if battery power is OK and CMOS checksum is OK.*
  - *Verify CMOS checksum manually by reading storage area*
- *Cryptographic library functionality test, and*
  - *HMAC-SHA-256 integrity check of the library*
  - *HMAC-SHA-1 KAT*
  - *HMAC-SHA-256 KAT*
  - *HMAC-SHA-384 KAT*
  - *AES 128 ECB Encrypt and Decrypt KAT*
  - *AES 256 GCM Encrypt and Decrypt KAT*
  - *RSA 2048 SHA-256 Sign and Verify KAT*
  - *ECDSA P-224 SHA-512 Sign and Verify PCT*
  - *DRBG AES-CTR-256 KAT (invoking the instantiate, reseed, and generate functions)*
- *Firmware integrity checks*
  - *RSA 2048 SHA-512 digital signature verification of the manifest file. This file contains a list of all executables that are part of the TSF*
  - *SHA-256 integrity check of each executable file in the TSF using the pre-calculated hashes from the manifest file.*].

5.2.5.5 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 [digital signature] prior to installing those updates.

5.2.6 TOE Access (FTA)

5.2.6.1 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

5.2.6.2 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.*

5.2.6.3 FTA_SSL.4 User-initiated Termination

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

5.2.6.4 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.

5.2.7 Trusted Path/Channels (FTP)

5.2.7.1 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 communications]*.

5.2.7.2 FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin
The TSF shall **be capable of using [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 contains 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 which is derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 13 below.

Table 13 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition

Assurance Class	Assurance Components	Component Description
	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 Ivanti to satisfy the assurance requirements. The following table lists the details.

Table 14 – 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 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.

SAR Component	How the SAR will be met
	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 15 – TOE Summary Specification SFR Description

Requirement	TSS Description
FAU_GEN.1	The TSF generates audit records for the following events: • Startup and shutdown of the audit function • Administrative login and logout events • Changes to TSF data related to configuration changes • Generation of a CSR and associated keypair • Installation of a certificate • Resetting passwords • Low audit storage space available • Failure to establish a HTTPS/TLS session • Failure to establish a TLS session • All use of the identification and authentication mechanism (local and remote connections to the TSF) • Unsuccessful attempts to validate a certificate • Initiation of a software update • Result of a software update • Changes to the time • Modification of the behavior of the TSF • Failure of self-tests • Initiation and termination of the trusted channel • Initiation and termination of the trusted path • Attempts to unlock an interactive session • Termination of a session by the session locking mechanism • ‘Trust issue’ with the certificate, e.g. due to failed path validation • Use of an ‘expired certificate’ • Absence of basicConstraints extension • CA flag not set for a certificate presented as a CA • Signature validation failure for any certificate in the certificate path; failure to establish revocation status; revoked certificate Each audit record includes the date and time, type, subject identity (IP address, hostname, and/or username), the outcome (success or failure), and any additional information specified in column three of Table 12. Certificates are identified in the log by the Certificate DN. All generating/importing of changing or deleting of cryptographic keys relate to certificates. Public keys associated with certificates are identified by the certificate DN and the term ‘public key’.
FAU_GEN.2	None
FAU_STG.1	The TOE is a standalone TOE. By default, the TSF allocates 200 MB to local audit storage; however, the administrator can configure the file size, up to 500 MB. The TSF divides the local audit storage between two audit files (active and inactive). When the current audit file reaches capacity; the TSF overwrites the inactive log file (if present). If the inactive log file is not present, then the TOE creates a new log file,

Requirement	TSS Description
	switches logging to the new log file, and generates an audit log indicating that a log file reached capacity. The TSF protects audit data from unauthorized modification and deletion though the restrictive administrative interfaces. The filesystem of the TSF is not exposed to the administrative user over the HTTPs GUI or the local CLI. The administrative user must be positively identified and authenticated prior to being allowed to clear the local audit log or change audit settings. The TSF implements Syslog over TLS using either TLS v1.1 or TLS v1.2. Logs are sent to the Syslog servers is real-time. The logs are also stored locally in case the connection to the remote syslog server cannot be established. The trusted channel with the Syslog server is described in greater detail in the FCS_TLSC_EXT.2 description.
FAU_STG_EXT.1	The TOE is a standalone TOE. By default, the TSF allocates 200 MB to local audit storage; however, the administrator can configure the file size, up to 500 MB. The TSF divides the local audit storage between two audit files (active and inactive). When the current audit file reaches capacity; the TSF overwrites the inactive log file (if present). If the inactive log file is not present, then the TOE creates a new log file, switches logging to the new log file, and generates an audit log indicating that a log file reached capacity. The TSF protects audit data from unauthorized modification and deletion though the restrictive administrative interfaces. The filesystem of the TSF is not exposed to the administrative user over the HTTPs GUI or the local CLI. The administrative user must be positively identified and authenticated prior to being allowed to clear the local audit log or change audit settings. The TSF implements Syslog over TLS using either TLS v1.1 or TLS v1.2. Logs are sent to the Syslog servers is real-time. The logs are also stored locally in case the connection to the remote syslog server cannot be established. The trusted channel with the Syslog server is described in greater detail in the FCS_TLSC_EXT.2 description.
FCS_CKM.1	The TSF supports the generation of RSA 2048 bit and 3072 bit keys for TLS client authentication, TLS server authentication, and RSA key encapsulation. The TSF generates ECDSA P-256 and P-384 keys for TLS client authentication, TLS server authentication, and TLS ECDHE key establishment.

FCS_CKM.2	The TSF uses both elliptic curve-based and RSA-based key establishment in support of TLS. When the TOE is configured with a server certificate with an RSA key, then RSA-based key establishment is used and the TOE acts as the sender. When the TOE is configured with a server certificate with an ECDSA key, then elliptic curve-based establishment is used and the TOE acts as the sender. The TOE supports the following schemes for key establishment: • RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, “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”

Requirement	TSS Description
	For syslog server, the TSF acts as the client and is the recipient. For these sessions, the TSF utilizes elliptic curve key agreement when an ECDHE TLS ciphersuite is negotiated and RSA based key encapsulation when any other TLS ciphersuite is negotiated. See FCS_TLS* description below for more details.
FCS_CKM.4	The TSF stores the following persistent keys on internal Hard Disk Drives in plaintext: • HTTPs/TLS Private Host Key – generated using the DRBG and FCS_CKM.1 or entered by the Security Administrator. • Syslog/TLS Private Client Key – generated using the DRBG and FCS_CKM.1 or entered by the Security Administrator. The HTTPS/TLS Private Host Key and the Syslog/TLS Private Client key are zeroized from the disk when the Security Administrator deletes the key, replaces the key, or zeroizes the entire TOE. The TSF zeroizes the HTTPs/TLS Private Host Key and the Syslog/TLS Private Client key on the hard disk drives by overwriting the file location with data from /dev/random three times. Each overwrite calls /dev/random ensuring that a different pseudo random pattern is used each time. The TSF stores loads the persistent keys into RAM when they are used and the TSF also stores the following ephemeral keys in RAM: • TLS Session keys – Established according to FCS_CKM.2 and derived using the TLS KDF • DRBG State – Derived from the entropy source HTTPS/TLS keys are zeroizedfrom RAM when the HTTP or Syslog process terminates. The TLS Session keys are zeroized from RAM when the associated TLS session is terminated. The DRBG state and all ephemeral keys are zeroized when the TSF is shutdown, suffers loss of power, or restarted. The TSF zeroizes keys in RAM by writing zeros to the memory location one time and performing a read verify to ensure that the memory location was set to all zeros. If the read verify fails, the TSF repeats thezeroization process. The above key destruction methods apply to all configurations and circumstances, except one. The only situation where the key destruction may be prevented would be if the system suffers a crash or loss of power. This situation only impacts the keys that are stored on the disk. Since the TOE is inaccessible in this situation, administrative zeroization cannot be performed. However, all keys on the disk are protected because the TOE enables full disk encryption by default. For additional details on Cryptographic Key Destruction, please refer to Section **Error! Reference source not found.**.
FCS_COP.1/DataEncryption	The TOE provides AES encryption/decryption in CBC and GCM modes with 128- and 256-bit keys.

FCS_COP.1/Hash	The TOE provides cryptographic hashing services for key generation using SHA-256 as specified in NIST SP 800-90 DRBG. SHA-1, SHA-256, and SHA-384 are used in support of TLS. SHA-256 is used for file integrity checking and password obfuscation. SHA-512 is used for hashing of the digital signature to verify the firmware manifest file.
FCS_COP.1/KeyedHash	The TOE implements HMAC message authentication for the following uses: • TLSv1.1 Master Secret Derivation: HMAC-SHA1, key sizes of 128 bits with ECDH P-256 or 192 bits with RSA and ECDH P-384, block size 512 bits, and output length of 160 bits;

Requirement	TSS Description
	• TLSv1.2 Master Secret Derivation: HMAC-SHA256, key sizes of 128 bits with ECDH P-256 or 192 bits with RSA and ECDH P-384, block size 512 bits, and output length of 256 bits; • TLSv1.2 Master Secret Derivation: HMAC-SHA384, key sizes of 256 bits with ECDH P-256 or 384 bits with RSA and ECDH P-384, block size 1024 bits, and output length of 384 bits; • TLSv1.1 Key Block Derivation: HMAC-SHA1, key size of 192 bits, block size of 512 bits, and output length of 160 bits; • TLSv1.2 Key Block Derivation: HMAC-SHA256, key size of 384 bits, block size of 512 bits, and output length of 256 bits; • TLSv1.2 Key Block Derivation: HMAC-SHA384, key size of 384 bits, block size of 1024 bits, and output length of 384 bits • TLS Message Authentication: HMAC-SHA1, key size of 160 bits, block size 512 bits, and output length of 160 bits
FCS_COP.1/SigGen	The TOE supports signature generation and verification with RSA (2048-. 3072-bit) with SHA-1/256/384/512 in accordance with FIPS PUB 186-4 and ECDSA with NIST curves P-256 and P-384 with SHA-1/256/384/512 in accordance with FIPS PUB 196-4. These signatures support TLS authentication.
FCS_HTTPS_EXT.1	The TSF implements the server and client sides of the HTTPs protocol according to RFC 2818 by using a TLS session to secure the HTTP session. All MUST and REQUIRED statement within RFC 2818 are followed. The TSF supports TLSv1.1 and TLSv1.2 for HTTPs/TLS. If the TSF receives a ClientHello message that requests TLSv1.0 or earlier, the TSF sends a fatal handshake_failure message and terminates the connection. When configured with an RSA certificate, the TSF supports the following TLS ciphersuties for connections to the TOE: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 When configured with an ECDSA certificate, the TSF supports the following TLS ciphersuites for connections to the TOE: • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 The TOE conforms to RFC 5246, section 7.4.3 for key exchange. When the TSF selects an ECDHE ciphersuite, it sends the client secp256r1 or secp384r1 key agreement parameters. The TSF prefers

Requirement	TSS Description
	secp256r1 if the client indicates support for both curves in the ClientHello message.
FCS_RBG_EXT.1	The TOE implements a DRBG in accordance with ISO/IEC 18031:2011 using a CTR DRBG with AES. The TSF seed the CTR_DRBG using 256-bits of data that contains at least 256 bits of entropy. The TSF gathers and pools entropy from 3 software based noise sources. • Device Specific randomness • Input layer timing randomness • Interrupt randomness The entropy sources are discussed in greater detail in the Entropy documentation.
FCS_TLSC_EXT.1 FCS_TLSC_EXT.2	The TSF implements a TLSv1.1 and TLSv1.2 client to secure communications with the Syslog server. The TSF supports and proposes the following ciphersuites and extensions in the ClientHello Message: • Ciphersuites for Syslog communications (FCS_TLSC_EXT.2): ○ TLS_RSA_WITH_AES_128_CBC_SHA ○ TLS_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA ○ TLS_RSA_WITH_AES_128_CBC_SHA256 ○ TLS_RSA_WITH_AES_256_CBC_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • Signature Algorithms: ○ RSA with SHA-1/256/384/512 ○ ECDSA SHA-1/256/384 • Supported Elliptic Curves: ○ secp256r1 ○ secp384r1 • Supported Point Formats ○ Uncompressed The Security Administrator can configure the TSF so it only accepts TLSv1.2 connections for server and client communications. The Security Administrator can enable and disable individual ciphersuites as well as specifying the preferred ordering of ciphersuites. The TOE sends the

Requirement	TSS Description
	supported elliptic curves extension if an ECDHE ciphersuite is selected and does not require administrator intervention. The TSF supports TLS with mutual authentication using X509v3 certificates to secure communications with the Syslog server. When the Syslog server sends the Certificate Request message, the TSFreplies with a Client Certificate message. The Client Certificate message includes the certificate that the Security Administrator configured to authenticate to the Syslog server. The TSF establishes reference identifiers for the remote server as follows: • When the server is specified using a domain name, the TSF verifies that the domain name matches a Subject Alternative Name DNS Name field in the certificate using exact or wildcard matching specified in Section 3.1 of RFC 2818. If the certificate does not contain any Subject Alternative Name fields, the TSF matches the domain name against the Common Name in the certificate. • When the server is specified using an IP address, the TSF verifies that the IP address exactly matches a Subject Alterative Name IP Address field in the certificate using the rules specified in Section 3.1 of RFC 2818. If the certificate does not contain any Subject Alternative Name fields, the TSF matches the IP address against the Common Name in the certificate. When the reference identifier is an IP address, the TOE converts the IP address to a binary representation in network byte order. IPv4 addresses are converted directly from decimal to binary as specified in RFC 3986. The TOE compares the binary IP address against all the IP Address entries in the Subject Alternative Name. The TSF does support wildcards but does not support certificate pinning and determines if the certificate is valid for the specified server based on the DNS name or IP address of the server. Wildcards are supported only at the left-most label of the identifier. In either instance, the TSF will not establish the connection if the peer certificate does not successfully authenticate the peer according to X.509 authentication. When acting as a server, the TSF listens on port 443 for HTTPs connections. The TSF uses HTML over HTTPs to present the administrative users with a secure management interface. The TSF uses TLS to provide a secure connection between the TSF and remote Security Administrators.

FCS_TLSS_EXT.1	The TSF supports TLSv1.1 and TLSv1.2 for HTTPs/TLS. If the TSF receives a ClientHello message that requests TLSv1.0 or earlier, the TSF sends a fatal handshake_failure message and terminates the connection. When configured with an RSA certificate, the TSF supports the following TLS ciphersuties for connections to the TOE: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256

Requirement	TSS Description
	- TLS_RSA_WITH_AES_256_CBC_SHA256 - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 When configured with an ECDSA certificate, the TSF supports the following TLS ciphersuites for connections to the TOE: - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 The TOE conforms to RFC 5246, section 7.4.3 for key exchange. When the TSF selects an ECDHE ciphersuite, it sends the client secp256r1 or secp384r1 key agreement parameters. The TSF prefers secp256r1 if the client indicates support for both curves in the ClientHello message. When acting as a server, the TSF listens on port 443 for HTTPs connections. The TSF uses HTML over HTTPs to present the administrative users with a secure management interface. The TSF uses TLS to provide a secure connection between the TSF and remote Security Administrators. The TOE does not support session resumption based on sessionID or session tickets.
FIA_AFL.1	An administrator can configure the number of unsuccessful attempts a remote administrator can make before a lock-out and can configure the length of time that the remote administrator is locked out. The attempts can be configured for a range between 3 and 10. The length of time can be configured between 10 and 999 minutes. If the user enters an incorrect password the configured number of times, the user is locked out they cannot login through any remote interface on the TOE. When the lockout time has expired, the administrator is allowed to authenticate to the TOE again. Lockouts are not enforced on the TOE’s console interface. This ensures that authentication failures cannot lead to a situation where no administrator access is available.
FIA_PMG_EXT.1	The TSF supports administrator password composition to include any combination of upper and lower case letters, numbers, and the following special characters “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, and the complete set of standard printable ASCII characters (values 0x20 – 0x7E)with a minimum length settable by the administrator and support 15 characters.

Requirement	TSS Description
FIA_UIA_EXT.1	The TSF utilizes HTTPS to secure a remote administration web UI session. When connecting over HTTPS, the TSF presents Security Administrators with a username and password prompt; The Security Administrator using password authentication is considered authenticated if the username and the SHA-256 hash of the password matches the stored username and SHA-256 password hash. A successful authentication takes the user to the System Status page. The TSF utilizes a local serial CLI which presents Security Administrators with a username and password prompt. The Security Administrator is considered authenticated if the username and password provided match the credentials configured in the TSF. A successful login takes the user to the CLI menu. Prior to successful identification and authentication, the TSF displays the TOE access banner specified in FTA_TAB.1 and responds to ICMP Echo messages with ICMP Echo Reply messages.
FIA_UAU_EXT.2	None
FIA_UAU.7	When the user is entering their password over the local console, the TSF does not echo any characters back.
FIA_X509_EXT.1/Rev	When a certificate is used (to identify the TSF or identify an external entity to the TSF), the TSF verifies certificates by checking the following: 1. The current date between the “Valid from” and “Valid to” dates. 2. The certificate is not listed on the CRL. If the TSF has a cached response that has not expired, the TSF uses the cached response in lieu of querying the CRL server. 3. The certificate chain is valid: • Each certificate in the certificate chain passes the checks described in #1 and #2. • Each certificate (other than the first certificate) in the certificate chain has the Subject Type=CA flag set. • Each certificate is signed by: ○ a certificate in the certificate chain, or ○ a trusted root CA that has been installed in the TSF The TSF verifies the validity of a certificate when: • An HTTPS client establishes a TLS connection (HTTPs Server Certificate) • The TSF verifies the server certificate of the Syslog server • The TSF uses its client certificate authenticate to the Syslog server If the Security Administrator loads a certificate with a Subject Type=CA, the TSF does not validate the certificate path. The rules for extendedKeyUsage fields are followed in all instances; Server Authentication purpose is checked for all presented Server certificates.
FIA_X509_EXT.2	When establishing a connection to the Syslog server, the TSF uses the certificate presented by the Syslog server to verify the server’s identity.

Requirement	TSS Description
	The TSF establishes reference identifiers for the remote server as follows: • When the server is specified using a domain name, the TSF verifies that the domain name matches a Subject Alternative Name DNS Name field in the certificate using exact or wildcard matching specified in Section 3.1 of RFC 2818. If the certificate does not contain any Subject Alternative Name fields, the TSF matches the domain name against the Common Name in the certificate. • When the server is specified using an IP address, the TSF verifies that the IP address exactly matches a Subject Alterative Name IP Address field in the certificate using the rules specified in Section 3.1 of RFC 2818. If the certificate does not contain any Subject Alternative Name fields, the TSF matches the IP address against the Common Name in the certificate. Once the TSF has verified that the certificate identifiers are valid for the Syslog server, the TSF verifies the validity of the certificate as described in FIA_X509_EXT.1/Rev. The TSF presents its own certificate to the Syslog server. This certificate is configured specifically for authentication to the Syslog server by the Security Administrator. If the TSF cannot contact the CRL server or the server does not respond, the TSF logs the failure and considers the certificate valid. If any of the other FIA_X509_EXT.1/Rev validity checks fail, the TSF rejects the certificate and does not establish the connection.
FIA_X509_EXT.3	The TSF allows Security Administrators to generate Certificate Signing Requests. The TSF requires the Security Administrator to specify the following values: • Common Name • Organization • Locality • State • Country • Key Type (RSA or ECDSA) • Key Length (2048, 3072, P-256, or P-384) The TSF allows the Security Administrator to specify an Organization Unit and additional random data used when generating the key pair. This information is optional for creating Certificate Signing Requests.
FMT_MOF.1/Functions FMT_MOF.1/ManualUpdate	The TSF implements the Security Administrator role to authorized administrators of the TOE. The TSF allows the Security Administrators to administer the TSF via CLI through a serial cabled connected to the TOE and a web UI over a remote HTTPS channel. The TSF permissions restrict access to these management functions to users that have been identified, authenticated, and authorized with the Security

Requirement	TSS Description
	Administrator role. The web UI and local console allow the Security Administrator to perform the following TSF management functions: • Verify/Install Firmware Updates • View/Edit settings for sending audit data to the Syslog Server • View/Edit the amount of space allocated Local Audit storage • Clear/Delete Local Audit records • View/Edit enabled TLS versions • View/Edit enabled TLS ciphersuties • View/Edit X.509 Certificates • Generate and configure cryptographic keys used to identify the TOE • Configure cryptographic keys used to authenticate users • View/Edit the TOE access banner • View/Edit the session inactivity timeout • View/Edit authentication failure parameters • Set user account passwords • Modify system time The administrative interfaces provided by the TSF do not allow any of these functions to be accessed by unauthenticated or unauthorized users. The TOE provides a trust store to store certificates. The permissions on the trust store restrict access so that only Security Administrators can import or delete certificates from the trust store. Security Administrators can also view the certificates stored in the trust store. No other access to the trust store is allowed.
FMT_MTD.1/CoreData	The administrative interfaces provided by the TSF do not allow any of the functions to be accessed by unauthenticated or unauthorized users. The TOE provides a trust store to store certificates. The permissions on the trust store restrict access so that only Security Administrators can import or delete certificates from the trust store. Security Administrators can also view the certificates stored in the trust store. No other access to the trust store is allowed. The only functions accessible prior to authentication are the display of the configurable warning and consent banner and the automated response to ICMP echo messages with ICMP echo reply messages.
FMT_MTD.1/CryptoKeys	The TOE restricts the ability to manage TLS (session keys), and any configured X.509 certificates (public and private key pairs) to security administrators via GUI.

FMT_SMF.1	The TSF implements the Security Administrator role to authorized administrators of the TOE. The TSF allows the Security Administrators to administer the TSF via CLI through a serial cabled connected to the

Requirement	TSS Description
	TOE and a web UI over a remote HTTPS channel. The TSF permissions restrict access to these management functions to users that have been identified, authenticated, and authorized with the Security Administrator role. The web UI and local console allow the Security Administrator to perform the following TSF management functions: • Verify/Install Firmware Updates • View/Edit settings for sending audit data to the Syslog Server • View/Edit the amount of space allocated Local Audit storage • Clear/Delete Local Audit records • View/Edit enabled TLS versions • View/Edit enabled TLS cipher suites • View/Edit X.509 Certificates • Generate and configure cryptographic keys used to identify the TOE • Configure cryptographic keys used to authenticate users • View/Edit the TOE access banner • View/Edit the session inactivity timeout • View/Edit authentication failure parameters • Set user account passwords • Modify system time • Configure audit behavior • Modify the behavior of the transmission of audit data to an external IT entity. The local console allows the Security Administrator to perform the following TSF management functions: • Verify firmware update • Set user account passwords The administrative interfaces provided by the TSF do not allow any of these functions to be accessed by unauthenticated or unauthorized users.
FMT_SMR.2	The following user roles are supported by the TOE: • System Administrator : Who have full read and write access to Admin UI pages. • Read-only-Administrators : System Administrator can enable these roles whose write access is restricted and can only read admin UI • Delegate Administrators : System administrator can create role with some endpoints read or write access.
FPT_APW_EXT.1	The TSF does not store plaintext password. The TSF stores the SHA-256 hash of each users' password. Additionally, the TSF does not provide a user interface to view the password hashes.

FPT_SKP_EXT.1	The TSF stores pre-shared keys, symmetric keys, and private keys in plaintext on the hard disk; however, it does not provide an interface to allow any user to view any of these values.
FPT_STM_EXT.1	The TOE time function is reliant on the system clock provided by the underlying hardware. The time source is maintained by a reliable

Requirement	TSS Description
	hardware clock that is updated by a Security Administrator once a month. The TSF uses system time to timestamp audit log records, to determine user session timeouts, and to determine certificate validity. These uses of time do not require an accuracy finer that one second, and the frequency of updating the time keeps the clock drift under one second. For virtual appliance, the TOE provides an ability to set the time manually.
FPT_TST_EXT.1	The TSF performs the following hardware self-tests at power-on: • BIOS checks at power-on (on hardware platforms only) ○ Verify boot block checksum. System will hang here if checksum is bad. ○ Verify main BIOS checksum. ○ Check CMOS diagnostic byte to determine if battery power is OK and CMOS checksum is OK. ○ Verify CMOS checksum manually by reading storage area. If the CMOS checksum is bad, update CMOS with power-on default values and clear passwords. • Cryptographic library tests: ○ HMAC-SHA-256 integrity check of the library ○ HMAC-SHA-1 KAT ○ HMAC-SHA-256 KAT ○ HMAC-SHA-384 KAT ○ AES 128 ECB Encrypt and Decrypt KAT ○ AES 256 GCM Encrypt and Decrypt KAT ○ RSA 2048 SHA-256 Sign and Verify KAT ○ ECDSA P-224 SHA-512 Sign and Verify PCT ○ DRBG AES-CTR-256 KAT (invoking the instantiate, reseed, and generate functions) • Firmware checks: ○ RSA 2048 SHA-512 digital signature verification of the manifest file. This file contains a list of all executables that are part of the TSF ○ SHA-256 integrity check of each executable file in the TSF using the pre-calculated hashes from the manifest file. The BIOS checks the successful use of the hardware platform to perform cryptographic operations and provides basic assurance that the hardware is working properly. The Cryptographic library test and the Firmware checks provide a high level of assurance that the firmware has not been tampered with and that the cryptographic algorithms are working properly. The Cryptographic library tests verify that each cryptographic algorithm[1] specified in FCS_COP.1 requirements is passing a KAT. The KAT demonstrates that the algorithm is functioning properly by invoking the algorithm with hard coded keys and messages and comparing the result to a pre-computed, known to be correct value. The

Requirement	TSS Description
	ECDSA PCT shows that the ECDSA algorithm is functioning properly by signing a known value with a known key and verifying that verifying the computed signature indicates that the signature is valid. If the BIOS checks fail, the TSF does not power-up. If the cryptographic library tests fail, the TSF will not start up. If any of the other checks fail, the TSF will log the failure and continue to boot.
FPT_TUD_EXT.1	The TSF allows the Security Administrator to install firmware updates. The Security Administrator obtains candidate updates by downloading them from the Ivanti Secure website. When the Security Administrator uploads a firmware update, the TSF performs an RSA 2048 SHA-256 digital signature verification of the update using the Ivanti Secure firmware update public key. The public key is distributed as part of the firmware package. Ivanti Secure retains control over the private key used to sign firmware updates. If the signature check is successful, the TSF installs the update. If the signature check detects tampering with the update and/or signature, the TSF presents the userwith an error message and discards the update. The TSF allows the Security Administrator to view the currently running version of firmware from the System Maintenance > Platform page of the web UI.
FTA_SSL.3	User sessions can be terminated by users. The Security Administrator can set the TOE so that local and remote sessions are terminated after a Security Administrator-configured period of inactivity.
FTA_SSL.4	User sessions can be terminated by users. The Security Administrator can set the TOE so that local and remote sessions are terminated after a Security Administrator-configured period of inactivity.
FTA_SSL_EXT.1	User sessions can be terminated by users. The Security Administrator can set the TOE so that local and remote sessions are terminated after a Security Administrator-configured period of inactivity.
FTA_TAB.1	The TSF enables Security Administrators to configure an access banner provided with the authentication prompt. The banner can provide warnings against unauthorized access to the TOE as well as any other information that the Security Administrator wishes to communicate. The TSF presents the access banner prior to authentication when a user connects to the remote web UI or local console CLI described in the FIA_UIA_EXT.1, FIA_UAU_EXT.2 description.
FTP_ITC.1	The TSF communicates with the external syslog server using Syslog over TLS with Authentication as described in the descriptions of FAU_STG_EXT.1 and FCS_TLSC_EXT.2. The TSF initiates the trusted channel with the Syslog server.
FTP_TRP.1/Admin	The TSF provides a trusted path for remote administration using HTTPs/TLS as described in FCS_HTTPS_EXT.1 and FCS_TLSS_EXT.1 descriptions.

6.1 CAVP Algorithm Certificate Details

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

Table 16 – CAVP Algorithm Certificate References

SFR	Algorithm	Key Size	Description	Certificate
FCS_CKM.1	RSA KeyGen (FIPS186-4)	2048,3072	Key Generation	A3010
	ECDSA KeyGen (FIPS186-4)	P-256 P-384	Key Generation	A3010
FCS_CKM.2	RSA (RFC 3447)	N/A	Key Establishment	No NIST CAVP certificate.
	KAS-ECC-SSC (Sp800-56Ar3)	P-256 P-384		A3010
FCS_COP.1/DataEncryption	AES-CBC AES-GCM	128,256	Encryption/Decryption	A3010
FCS_COP.1 /SigGen and SigVer	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	2048,3072	Signature Generation and Verification	A3010
	ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)	P-256, P-384		
FCS_COP.1/Hash	SHA-1 SHA-256 SHA-384 SHA-512	160, 256, 384, 512	Hashing	A3010
FCS_COP.1/KeyedHash	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384	160, 256, 384	Keyed-Hashing	A3010
FCS_RBG_EXT.1	Counter DRBG (AES-256)	-	Random Bit Generation	A3010

6.2 Cryptographic Key Destruction

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

Table 17 – Zeroization

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
HTTPs/TLS Private Host Key	Generated using the DRBG and FCS_CKM.1 or entered by the Security Administrator.	Persistent keys on internal Hard Disk Drives in plaintext	The HTTPS/TLS Private Host Key and the Syslog/TLS Private Client key are zeroized from the disk when the Security Administrator deletes the key, replaces the key, or zeroizes the entire TOE. These keys are zeroized from RAM when the HTTP or Syslog process terminates, the TSF is shutdown, suffers loss of power, or restarted.
Syslog/TLS Private Client Key	Generated using the DRBG and FCS_CKM.1 or entered by the Security Administrator.	Persistent keys on internal Hard Disk Drives in plaintext	The HTTPS/TLS Private Host Key and the Syslog/TLS Private Client key are zeroized from the disk when the Security Administrator deletes the key, replaces the key, or zeroizes the entire TOE. These keys are zeroized from RAM when the HTTP or Syslog process terminates, the TSF is shutdown, suffers loss of power, or restarted.
TLS Session keys	Established according to FCS_CKM.2 and derived using the TLS KDF	The TSF stores loads the persistent keys into RAM when they are used and the TSF also stores the following ephemeral keys in RAM	The TLS Session keys are zeroized from RAM when the associated TLS session is terminated, the TSF is shutdown, suffers loss of power, or restarted.
DRBG State	Derived from the entropy source	The TSF stores loads the persistent keys into RAM when they are used and the TSF also stores the following ephemeral keys in RAM	The DRBG state is zeroized when the TSF is shutdown, suffers loss of power, or restarted.

7 Acronym Table

Acronyms should be included as an Appendix in each document.

Table 18 – 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
OSP	Organizational Security Policy
PP	Protection Profile
RSA	Rivest, Shamir & Adleman
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSF	TOE Security Functionality (TSF = TOE for pND or Case 1 vND according to section 1.2)
TSS	TOE Summary Specification