VMware Unified Access Gateway (UAG) 2209

Security Target

Version 1.0

24 April 2023

Prepared for:

VMware, Inc.
3401 Hillview Avenue
Palo Alto, CA 94304

Prepared by:

Accredited Testing and Evaluation Labs
6841 Benjamin Franklin Drive
Columbia, MD 21046

Contents

Tables

1 Security Target Introduction

The Security Target (ST) contains the following additional sections:

- Product and TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)

1.1 Security Target, TOE, and CC Identification

ST Title – VMware Unified Access Gateway (UAG) 2209 Security Target

ST Version – Version 1.0

ST Date – 24 April 2023

TOE Identification – VMware Unified Access Gateway (UAG) 2209

TOE Developer – VMware, Inc.

Evaluation Sponsor – VMware, Inc.

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- *collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020* (NDcPP) with the following optional and selection-based SFRs:
  - FAU_STG.1
  - FCS_HTTPS_EXT.1
  - FCS_TLSC_EXT.1
  - FCS_TLSS_EXT.1
  - FCS_TLSS_EXT.2
  - FIA_X509_EXT.1/Rev
  - FIA_X509_EXT.2
  - FIA_X509_EXT.3
  - FMT_MOF.1/Functions
  - FMT_MTD.1/CryptoKeys
- The following NIAP Technical Decisions apply to the TOE and have been accounted for in the ST development and the conduct of the evaluation, or were considered to be non-applicable:

  TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1)

  - This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4

- This TD is not applicable to the TOE; the TOE does not claim FCS_NTP_EXT.1.

TD0536: NIT Technical Decision for Update Verification Inconsistency

- This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

TD0537: NIT Technical Decision for Incorrect Reference to FCS_TLSC_EXT.2.3

- This TD is not applicable to the TOE. It applies to the evaluation of FCS_TLSC_EXT.2, which the TOE does not claim.

TD0538: NIT Technical Decision for Outdated Link to Allowed-With List

- This TD is not applicable to the TOE. The purpose of the TD was to provide an updated reference to the list of PP-Modules that may be included with the claimed PP. In addition to being a modification of the PP itself and not anything that affects a TOE that conforms to the PP, this TOE does not claim any PP-Modules, so it is not applicable to the TOE for that reason as well.

TD0546: NIT Technical Decision for DTLS – Clarification of Application Note 63

- This TD is not applicable to the TOE. The TD applies to DTLS requirements, which the TOE does not claim.

TD0547: NIT Technical Decision for Clarification on Developer Disclosure of AVA_VAN

- This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

TD0555: NIT Technical Decision for RFC Reference Incorrect in TLSS Test

- This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

TD0556: NIT Technical Decision for RFC 5077 Question

- This TD is not applicable to the TOE. The TOE does not claim conformance to RFC 5077.

TD0563: NIT Technical Decision for Clarification of Audit Date Information

- This TD is applicable to the TOE but applies specifically to an application note in the PP for an SFR that the TOE claims so the ST itself is unaffected.

TD0564: NIT Technical Decision for Vulnerability Analysis Search Criteria

- This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7

- This TD is applicable to the TOE.

TD0570: NIT Technical Decision for Clarification About FIA_AFL.1

- This TD is applicable to the TOE.

TD0571: NIT Technical Decision for Guidance on How to Handle FIA_AFL.1

- This TD is applicable to the TOE, but it only affects how a claimed SFR is interpreted so the ST itself is unaffected.

TD0572: NIT Technical Decision for Restricting FTP_ITC.1 to Only IP Address Identifiers

- This TD is applicable to the TOE, but it only affects how a claimed SFR is interpreted so the ST itself is unaffected.

TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e

- This TD is applicable to the TOE.

TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3

- This TD is applicable to the TOE.

TD0591: NIT Technical Decision for Virtual TOEs and Hypervisors

- This TD is applicable to the TOE but there is no change that directly affects the content of the ST.

TD0592: NIT Technical Decision for Local Storage of Audit Records

- This TD is applicable to the TOE but there is no change that directly affects the content of the ST.

TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server

- This TD is not applicable to the TOE. The TOE does not claim SSH server functionality.

TD0632: NIT Technical Decision for Consistency with Time Data for vNDs

- This TD is applicable to the TOE.

TD0633: NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance

- This TD is not applicable to the TOE. The TOE does not claim IPsec functionality.

TD0634: NIT Technical Decision for Clarification required for testing IPv6

- This TD is not applicable to the TOE. The TOE does not support IP address as a reference identifier for TLS server certificates.

TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters

- This TD is applicable to the TOE but there is no change that directly affects the contents of the ST.

TD0636: NIT Technical Decision for Public Key User Authentication for SSH

- This TD is not applicable to the TOE. The TOE does not claim SSH functionality.

TD0638: NIT Technical Decision for Key Pair Generation for Authentication

  - This TD is applicable to the TOE.

  TD0639: NIT Technical Decision for NTP MAC Keys

  - This TD is not applicable to the TOE. The TOE does not claim NTP functionality.

  TD0670: NIT Technical Decision for Mutual and Non-Mutual TLSC testing

  - This TD is applicable to the TOE.

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 5, April 2017.
  - Part 3 Conformant

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. An iterated SFR is indicated by adding a string starting with '/' (e.g. 'FCS_COP.1/Hash').
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using italics and are surrounded by brackets (e.g., [*assignment item*]). Note that an assignment within a selection would be identified in both italics and underline, with the brackets themselves underlined since they are explicitly part of the selection text, unlike the brackets around the selection itself (e.g., [selection item, [*assignment item inside selection*]]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using underlines and are surrounded by brackets (e.g., [selection item]).
  - Refinement: allows technical changes to a requirement to make it more restrictive and allows non-technical changes to grammar and formatting. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that minor grammatical changes that do not involve the addition or removal of entire words (e.g., for consistency of quantity such as changing "meets" to "meet") do not have formatting applied.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.
- The ST does not show selection/assignment operations that have been completed by the PP authors, though it does preserve brackets to show where such operations have been made.

1.3.1 Terminology

The following terms and abbreviations are used in this ST:

Table 1: Terms and Definitions

Term	Definition
Agent	A Horizon component that acts as an endpoint on a protected resource and serves content on that resource (individual applications or an interactive desktop session) to an authorized Horizon Client.
Blast	A communications protocol that is used to transmit interactive desktop and application sessions (user inputs and audio/visual outputs).
Client	A Horizon component that resides on an end user device that the user can run to access enterprise computing resources via the virtual desktop.
Cloud Pod	A self-contained Horizon deployment on a particular network. Multiple cloud pods can be federated, allowing a client on one pod to access resources on another.
Connection Server	A Horizon component that is responsible for determining the authorizations of a Horizon Client user and facilitating the establishment of Agent communications so that authorized resources can be served to that user.
Horizon	A collection of products that are used to allow an organizational user to access shared enterprise resources in a protected network from a single client application.
Unified Access Gateway	A network device that acts as a proxy between a Horizon Client on an unprotected network and other Horizon components on a protected internal network. The Unified Access Gateway is responsible for authenticating Horizon Client users and passing their validated identity to a Connection Server via SAML assertion. It is also responsible for establishing Horizon Agent connectivity on behalf of the client.
Virtual Desktop	The virtual desktop is the set of enterprise computing resources that are served to a user within an interactive Horizon Client session. For the purpose of the TSF, the important consideration is that all virtual desktop content is transmitted over TLS.

1.3.2 Acronyms

Table 2: Acronyms

Term	Definition
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
DTLS	Datagram Transport Layer Security
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
FQDN	Fully Qualified Domain Name
GCM	Galois Counter Mode
JVM	Java Virtual Machine

NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PBKDF	Password-Based Key Derivation Function
PP	Protection Profile
RBG	Random Bit Generator
rDSA	RSA Digital Signature Algorithm
RNG	Random Number Generation
SAML	Security Assertion Markup Language
SAN	Subject Alternative Name
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
UAG	Unified Access Gateway
UPN	User Principal Name
VPN	Virtual Private Network

2 Product and TOE Description

2.1 Introduction

The VMware Unified Access Gateway (UAG) is a virtual network device that is used as a remote access server to allow users on an untrusted network (e.g. a home office or other offsite location) to access enterprise resources on a protected internal network.

For this Security Target, the Target of Evaluation (TOE) is the UAG itself, which runs as a virtual network device on an environmental hypervisor and hardware platform, consistent with “Case 1” virtual network devices defined in section 1.2 of the NDcPP. For the tested configuration, the environmental hypervisor is VMware ESXi 7.0 and the environmental physical platform is a Dell PowerEdge R740 with an Intel Xeon 6230R (Cascade Lake) processor. The product has a variety of uses but the purpose of the TOE as evaluated is for facilitating connectivity between VMware Horizon users and virtual desktops/applications.

The TOE conforms to the NDcPP. As such, the security-relevant functionality of the product is limited to the claimed requirements in those standards. The security-relevant functionality is described in sections 2.3 and 2.4. The product overview in section 2.2 below is intended to provide the reader with an overall summary of the entire product so that its intended usage is clear. The subset of the product functionality that is within the evaluation scope is subsequently described in the sections that follow it.

2.2 Product Overview

The VMware UAG is a secure remote access gateway that acts as a reverse proxy for protected network resources and allows a user on an unprotected network to gain access to those resources.

The UAG is responsible for identification and authentication of remote users that are attempting to use other VMware products that are deployed within an enterprise environment. With respect to this Security Target, the primary use case for this is for VMware Horizon.

VMware Horizon is a suite of components that establish a virtualization environment within an organization. The Horizon product components collectively allow users to access virtualized desktops or enterprise resources from their end user device. These resources are made available with granular security controls that allow users to access only the capabilities for which they are authorized.

VMware Horizon as a suite consists of several components:

- Horizon Clients are applications that are installed on end user devices. A user accesses their virtual desktop through the Horizon Client.
- Horizon Agents are applications that run on virtual servers in the enterprise environment. These agents facilitate remote access to the desktop of a virtual server or to specific applications running on that server that may be served directly to the virtual desktop.
- The Horizon Connection Server is responsible for brokering connections between Horizon Clients and Horizon Agents to authenticate users and serve appropriate resources to a particular user based on enterprise permissions.

The UAG’s role in this is to be the initial gateway that a Horizon Client interacts with when attempting to access Horizon Agents. The UAG is responsible for maintaining a linkage between the external network connection initiated by the user and the internal network connection that it initiates to other Horizon components. The UAG authenticates the Horizon Client user and passes an assertion to the Connection Server that identifies the user. Based on the user’s privileges, the Connection Server notifies relevant

Horizon Agents that an authorized user is requesting access to them. The UAG then establishes a second connection to the relevant Agent(s) that is then used to pass interactions back and forth between the external Horizon Client and the internal Horizon Agent(s). All such interactions occur over separate TLS channels.

2.3 TOE Overview

The Target of Evaluation (TOE) is VMware UAG 2209. Specifically, the TOE is a virtual network device that includes its operating system (VMware Photon 3.0) and the software that runs on it. The TOE is a "Type 1" virtual network device as defined by section 1.2 of the NDcPP. The TOE boundary therefore includes only the virtualized network device, while its underlying hypervisor and physical platform are environmental.

With respect to the security functionality of the TOE, the TSF is limited to the relevant functionality that is defined in the claimed PP and package. The logical boundary of the TOE is summarized in section 2.4.2. However, the following general capabilities are considered to be within the scope of the TOE:

- **Protection of data in transit:** the TOE secures data in transit between itself and its operational environment using TLS and HTTPS.
- **Cryptographic services:** the TOE includes libraries with NIST-validated algorithm services that it uses for secure protocol implementations. It also provides X.509 certificate services in support of these protocols.
- **Identification, authentication, administration, and accounting:** the TOE includes mechanisms for authenticating remote and local administrators to facilitate authorized access to its management functions and recording the security-relevant actions that occur.
- **Self-protection:** the TOE includes various self-protection mechanisms to reduce the risk that the TSF or its data have their functionality altered through deliberate or accidental means.

2.4 TOE Architecture

The UAG TOE consists of a virtual network device running on an environmental hypervisor and physical platform. The TOE software includes VMware Photon 3.0, OpenSSL and Bouncy Castle BC-FJA cryptographic libraries, as well as specialized software needed to run the actual UAG functionality.

2.4.1 Physical Boundary

The TOE consists of the following component, as shown in Figure 1 below:

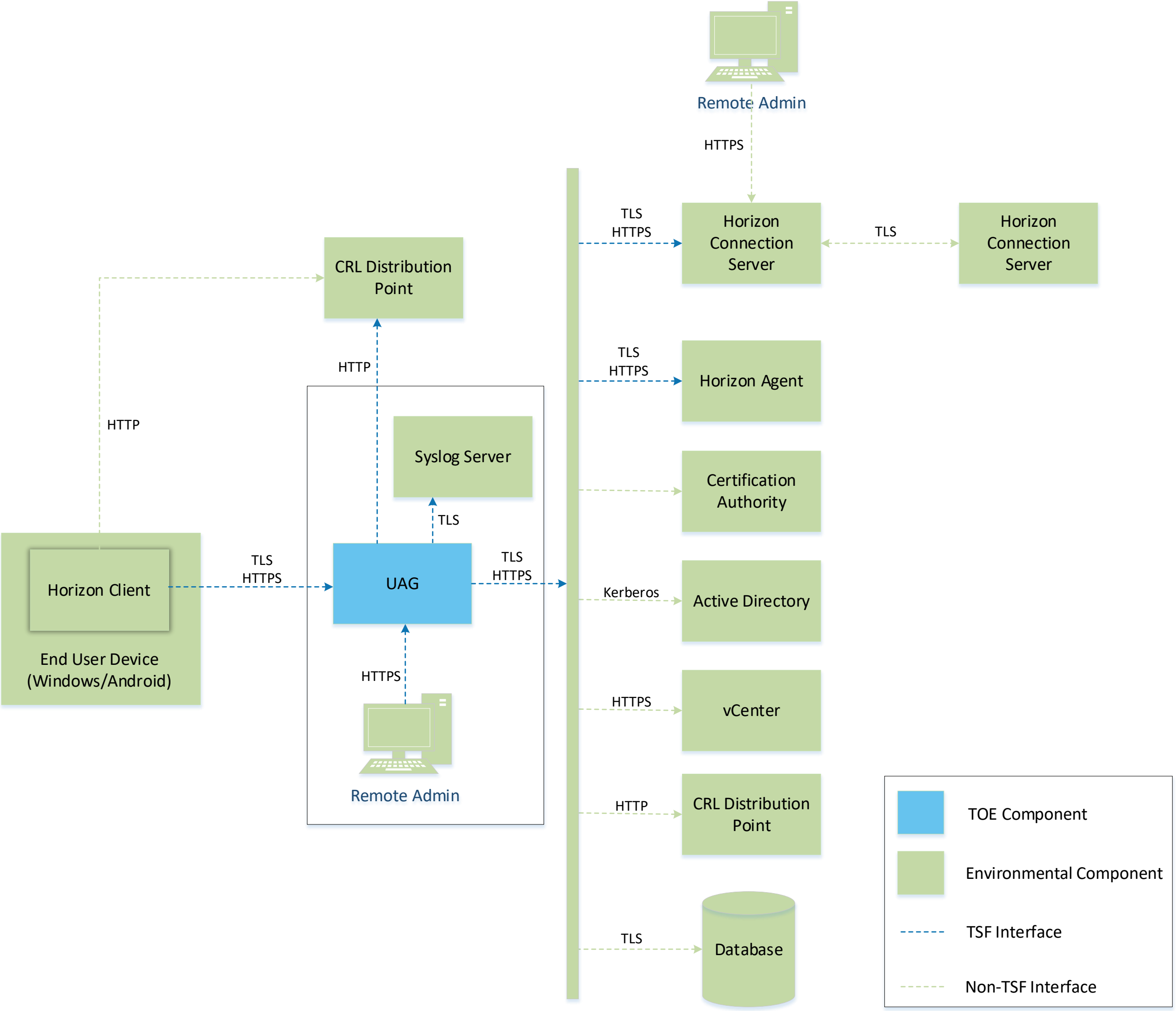
- VMware UAG 2209
    - Note that the specific tested version was "Unified Access Gateway (UAG) 2209.2 for vSphere (FIPS)" as this was the most recent release available at the time of testing – other versions of the UAG which include a non-FIPS version and a Microsoft Azure version were not tested

Figure 1 shows the TOE in a sample deployment with other VMware Horizon components in its operational environment. Note the following:

- Firewalls are not shown between internal and external networks but it is assumed that the TOE is deployed in a DMZ between them.

- Multiple UAGs may be deployed in a load balancer configuration to ensure resource availability. As the claimed PP does not have availability requirements, only one UAG is deployed in the tested configuration.
- The second Horizon Connection Server that is depicted on the diagram has its own associated Horizon Agents and other external interfaces. These are omitted for simplicity.
- The environment assumes that all components have access to the organization’s Certification Authority for issuance and validation of X.509 certificates.
- The ‘Database’ component refers to the optional Event Database (SQL Server or Postgres) that is used by the Connection Server if configured.

Figure 1 - TOE Boundary



The TOE handles inbound requests from the Horizon Client over both mutually-authenticated and one-way TLS. Initial Horizon Client connectivity to the TOE requires mutually-authenticated TLS but once authenticated, subsequent connections do not require re-validation of the client certificate. Inbound management communications and outbound communications, both to the environmental syslog server and to other Horizon components (Connection Server and Agent) use one-way TLS. While the TOE does not provide a certificate to a remote Horizon Agent for authentication, the Horizon Agent does require it to provide a single use authorization token that is issued to the TOE by the Connection Server.

The following network ports must be open for the TOE to function:

- TCP/443 (for inbound session establishment connectivity and Blast protocol connectivity from Horizon Clients)
- TCP/9443 (for inbound remote administration)

The TOE's operational environment includes the following:

- VMware Horizon components (at least one each of Horizon Client, Horizon Connection Server, and Horizon Agent)
- Remote syslog server
- Platform (hardware and software) on which the TOE is hosted. In the tested configuration, this included the following:
  - VMware ESXi 7.0
  - Dell PowerEdge R740 with Intel Xeon 6230R (Cascade Lake) processor
- Access to a Certification Authority and corresponding revocation checking mechanism is needed to validate presented X.509 certificates.
- A remote system with a supported web browser for remote administrative access:
  - The tested configuration used Google Chrome 106.0.5249.119

2.4.2 Logical Boundary

This section summarizes the security functions provided by the TOE:

- Security Audit
- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

2.4.2.1 Security Audit

The TOE generates audit records of security-relevant activity. Audit data is stored locally on the TOE in several different files based on event type; local audit records are protected against unauthorized modification and deletion. A log rotation exists to overwrite the oldest stored records when audit storage space has been exhausted. The TOE also has the ability to export all audit records to an external syslog server over a TLS protected channel.

2.4.2.2 Cryptographic Support

The TOE implements cryptographic functions in support of trusted communications, key pair generation for X.509 certificate requests, and self-testing. The TOE includes both OpenSSL and Bouncy Castle BC-FJA cryptographic libraries. For trusted communications, the TOE implements TLS as a server with HTTPS, and

TLS as a client with and without HTTPS. TLS/HTTPS server connectivity between the environmental Horizon Client and the TOE enforces mutual authentication of TLS client certificates. The TOE relies on platform hardware to generate entropy that is used to seed its DRBG to ensure that generated keys have the advertised security strength.

2.4.2.3 Identification and Authentication

The TOE uses a local password-based mechanism for administrator authentication. The TOE enforces restrictions on the length and character composition of administrator passwords. Excessive failed authentication attempts on a remote administrative interface will cause a lockout that is resolved by a waiting period. The TOE also uses X.509 certificates for authentication of TLS connections. The TOE has a mechanism by which a certificate signing request can be generated so that it may obtain a certificate for its own use from a trusted CA.

2.4.2.4 Security Management

The TOE has a web-based remote management interface as well as a local console. Most functionality is administered over the remote interface. The TOE uses a single Security Administrator role to authorize the use of management functions.

2.4.2.5 Protection of the TSF

The TOE protects sensitive data from unauthorized access. It enforces integrity of its own contents through the use of self-tests to ensure that the TSF has not been modified. Software updates are obtained through the operational environment (e.g. downloaded from the vendor’s support site); updates have a published hash that an administrator can verify prior to their application.

2.4.2.6 TOE Access

The TOE controls access through enforcement of idle session timeout on its management interfaces. These interfaces also display a configurable pre-authentication warning banner that advises against unauthorized use of the TOE.

2.4.2.7 Trusted Path/Channels

The TOE implements TLS and TLS/HTTPS trusted channels between itself and environmental systems. The TOE also implements a TLS/HTTPS trusted path for secure remote administration.

2.5 TOE Documentation

VMware provides the following product documentation in support of the installation and secure use of the TOE:

- Horizon Administration (https://docs.vmware.com/en/VMware-Horizon/2209/horizon-console-administration.pdf)
- Deploying and Configuring VMware Unified Access Gateway (https://docs.vmware.com/en/Unified-Access-Gateway/2209/uag-deploy-config-guide.pdf)
- VMware UAG Common Criteria Evaluated Configuration Guide

3 Security Problem Definition

This ST includes by reference the Security Problem Definition, composed of threats, assumptions, and organizational security policies from the NDcPP.

In general, the threat model of the NDcPP is designed to protect against the following:

- Unauthorized or insecure communications
- Invalid updates
- Undetected activity
- Unauthorized administrators
- Device failures

The NDcPP defines several assumptions that only apply to the TOE in certain circumstances; within the context of this ST, the A.COMPONENTS_RUNNING assumption does not apply because the TOE is a standalone device, and the A.VS_TRUSTED_ADMINISTRATOR, A.VS_REGULAR_UPDATES, and A.VS_ISOLATION assumptions all apply because the TOE is a virtual network device.

4 Security Objectives

Like the Security Problem Definition, this ST includes by reference the security objectives defined in the NDcPP. This includes only security objectives for the TOE's operational environment.

The NDcPP defines several objectives that only apply to the TOE in certain circumstances; within the context of this ST, the OE.COMPONENTS_RUNNING objective does not apply because the TOE is a standalone device, and the OE.VM_CONFIGURATION objective does apply because the TOE is a virtual network device.

5 IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the following Protection Profile (PP):

- *collaborative Protection Profile for Network Devices*, Version 2.2e, March 23, 2020

As a result, any selection, assignment, or refinement operations already performed by that PP on the claimed SFRs are not identified here (i.e., they are not formatted in accordance with the conventions specified in section 1.3 of this ST). Formatting conventions are only applied on SFR text that was chosen at the ST author's discretion.

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the NDcPP. These documents define the following extended SAR and extended SFRs; since they have not been redefined in this ST, the NDcPP should be consulted for more information regarding these extensions to CC Parts 2 and 3.

- FAU_STG_EXT.1
- FCS_HTTPS_EXT.1
- FCS_RBG_EXT.1
- FCS_TLSC_EXT.1
- FCS_TLSS_EXT.1
- FCS_TLSS_EXT.2
- FIA_PMG_EXT.1
- FIA_X509_EXT.1/Rev
- FIA_X509_EXT.2
- FIA_X509_EXT.3
- FIA_UAU_EXT.2
- FIA_UIA_EXT.1
- FPT_APW_EXT.1
- FPT_SKP_EXT.1
- FPT_STM_EXT.1
- FPT_TST_EXT.1
- FPT_TUD_EXT.1
- FTA_SSL_EXT.1

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Table 3: TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1 Audit Data Generation
	FAU_GEN.2 User Identity Association
	FAU_STG.1 Protected Audit Trail Storage

Requirement Class	Requirement Component
	FAU_STG_EXT.1 Protected Audit Event Storage
FCS: Cryptographic Support	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/Hash Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)
	FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)
	FCS_HTTPS_EXT.1 HTTPS Protocol
	FCS_RBG_EXT.1 Random Bit Generation
	FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication
	FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication
	FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication
FIA: Identification and authentication	FIA_AFL.1 Authentication Failure Management
	FIA_PMG_EXT.1 Password Management
	FIA_UAU.7 Protected Authentication Feedback
	FIA_UAU_EXT.2 Password-Based Authentication Mechanism
	FIA_UIA_EXT.1 User Identification and Authentication
	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: Security Management	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: Protection of the TSF	FPT_APW_EXT.1 Protection of Administrator Passwords
	FPT_SKP_EXT.1 Protection of TSF Data (for reading all pre-shared, symmetric, and private keys)
	FPT_STM_EXT.1 Reliable Time Stamps
	FPT_TST_EXT.1 TSF Testing
	FPT_TUD_EXT.1 Trusted Update
FTA: TOE Access	FTA_SSL.3 TSF-Initiated Termination
	FTA_SSL.4 User-Initiated Termination
	FTA_SSL_EXT.1 TSF-Initiated Session Locking

Requirement Class	Requirement Component
	FTA_TAB.1 Default TOE Access Banners
FTP: Trusted Path/Channels	FTP_ITC.1 Inter-TSF Trusted Channel
	FTP_TRP.1/Admin Trusted Path

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1 Audit Data Generation

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:

- Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- [no other actions]];

d) Specifically defined auditable events listed in Table **4**.

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 **4**.

Table 4: 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.

Requirement	Auditable Events	Additional Audit Record Contents
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_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure
FCS_TLSS_EXT.2	Failure to authenticate the client	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	Reason for failure of certificate validation.
	Any addition, replacement, or removal of trust anchors in the TOE's trust store	Identification of certificates added, replaced, or removed as trust anchor in the TOE's trust store.
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process.	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_TST_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism.	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_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions	None.

5.2.1.2 FAU_GEN.2 User Identity Association

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

5.2.1.3 FAU_STG.1 Protected Audit Trail Storage

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised 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: [*log file rotation*]] 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 keys in accordance with a specified cryptographic key generation algorithm: [

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
- ECC schemes using 'NIST curves' [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4
- FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 7919]].

Application Note: *The TOE uses RSA key generation for certificate signing requests (FIA_X509_EXT.3). The TOE uses ECC key generation for TLS server functionality (FCS_TLSS_EXT.1). The TOE uses both ECC and FFC key generation for TLS client functionality (FCS_TLSC_EXT.1). All TLS client interfaces use ECC key generation but only the TLS client interfaces that are implemented through BC-FJA support FFC key generation.*

5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment[1]

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

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"]
- FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 7919]].

Application Note: *The TOE uses ECC key establishment for TLS server functionality (FCS_TLSS_EXT.1). The TOE uses both ECC and FFC key establishment for TLS client functionality (FCS_TLSC_EXT.1).*

[1] Modified by 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 [destruction of reference to the key directly followed by a request for garbage collection];
- 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 [*instructs a part of the TSF to destroy the abstraction that represents the key*]

that meets the following: No Standard.

5.2.2.4 FCS_COP.1/DataEncryption Cryptographic Operation

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic 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/Hash Cryptographic Operation (Hash Algorithm) (TSF Self-Test)

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 message digest sizes [160, 256, 384, 512] bits that meet the following: ISO/IEC 10118-3:2004.

5.2.2.6 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 [*512 bits*] and message digest sizes [160, 256, 384] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.

5.2.2.7 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) [*3072 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].

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 HTTPS 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 Services

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

Application Note: *The TOE's OpenSSL cryptographic library uses CTR_DRBG and the TOE's Bouncy Castle cryptographic library uses Hash_DRBG.*

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[1] platform-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)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- 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.

Application Note: *All TLS client interfaces use TLS_ECDHE ciphersuites but only the TLS client interfaces that are implemented through BC-FJA support TLS_DHE ciphersuites.*

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [the reference identifier per RFC 6125 section 6].

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

FCS_TLSC_EXT.1.4 The TSF shall [present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: [secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups] in the Client Hello.

Application Note: *All TLS client interfaces support the claimed elliptic curves but only the TLS client interfaces that are implemented through BC-FJA support ffdhe groups.*

5.2.2.11 FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication

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

- 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 [TLS 1.1].

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [ECDHE curves [secp384r1] and no other curves]].

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

5.2.2.12 FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication

FCS_TLSS_EXT.2.1 The TSF shall support TLS communication with mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.2 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [

- Not implement any administrator override mechanism

].

FCS_TLSS_EXT.2.3 The TSF shall not establish a trusted channel if the identifier contained in a certificate does not match an expected identifier for the client. If the identifier is a Fully Qualified Domain Name (FQDN), then the TSF shall match the identifiers according to RFC 6125, otherwise the TSF shall parse the identifier from the certificate and match the identifier against the expected identifier of the client as described in the TSS.

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 [*1-100*] 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 [*locally-initiated unlock operation*] is taken by an Administrator, 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: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [*no other characters*]];

b) Minimum password length shall be configurable to between [*8*] and [*64*] characters.

5.2.3.3 FIA_UAU.7 Protected Authentication Feedback

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

5.2.3.4 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_UIA_EXT.1 User Identification and Authentication

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

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

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

5.2.3.6 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 updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

5.2.3.7 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 [TLS, HTTPS] and [no additional uses].

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

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] to Security Administrators.

5.2.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

FMT_MOF.1.1/ManualUpdate The TSF shall restrict the ability to enable the functions 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* [*hash comparison*] *capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1; [*
- *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
- *Ability to manage the cryptographic keys;*
- *Ability to re-enable an Administrator account;*
- *Ability to manage the TOE's trust store and designate X.509v3 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 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[2] 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 [obtain time from the underlying virtualization system].

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), periodically during normal operation] to demonstrate the correct operation of the TSF: [*cryptographic module self-test, file system integrity test, continuous RNG test, pairwise consistency test*].

Application Note: *Cryptographic module self-test and file system integrity tests are performed during start-up, other tests are performed conditionally as described in section 6.5.*

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 [published hash] prior to installing those updates.

[2] Modified by TD0632

5.2.6 TOE Access (FTA)

5.2.6.1 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.

5.2.6.2 FTA_SSL.4 User-Initiated Termination

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

5.2.6.3 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.

5.2.6.4 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, HTTPS] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [[*Horizon Client user authentication, Horizon Client media transmission, Horizon Connection Server session establishment, Horizon Agent media transmission*]] 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 [*remote audit storage, Horizon Connection Server session establishment, Horizon Agent media transmission*].

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 Security Assurance Requirements

The security assurance requirements for the TOE are included by reference to the NDcPP.

Table 5: Assurance Components

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1 Basic Functional Specification
AGD: Guidance Documentation	AGD_OPE.1 Operational User Guidance
	AGD_PRE.1 Preparative Procedures
ALC: Life-cycle Support	ALC_CMC.1 Labeling of the TOE
	ALC_CMS.1 TOE CM Coverage
ASE: Security Target	ASE_CCL.1 Conformance Claims
	ASE_ECD.1 Extended Components Definition
	ASE_INT.1 ST Introduction
	ASE_OBJ.1 Security Objectives for the Operational Environment
	ASE_REQ.1 Stated Security Requirements
	ASE_SPD.1 Security Problem Definition
	ASE_TSS.1 TOE Summary Specification
ATE: Tests	ATE_IND.1 Independent Testing – Conformance
AVA: Vulnerability Assessment	AVA_VAN.1 Vulnerability Survey

All SARs required by the NDcPP will apply to the entire TOE. The evaluation activities specified in the NDcPP apply to the entire TOE evaluation, including any changes made to them by subsequent NIAP Technical Decisions as summarized in section 1.2 above.

6 TOE Summary Specification

This chapter describes the security functions of the TOE:

- Security Audit
- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

6.1 Security Audit

The TOE is a standalone device that generates audit records that reside in its local storage. The TOE generates audit records of security-relevant management functions on both its local and remote interfaces. The audit records that are generated include startup and shutdown of the TOE (the audit functions are not enabled or disabled separately), all administrative actions, and the specific events identified in Table 4 above. For all auditable events that involve a user (e.g. authentication and administration events), the user identity is captured in the audit record. When key data is updated, the audit record also includes identifying information for the key (subject DN, issuer DN, thumbprint, and expiration). All audit records also include date and time of the event, type of event, subject identity, and the outcome of the event where appropriate.

Audit records are stored in several different log files. With respect to the TSF, the admin.log, audit.log, and esmanager.log files contain records of security-relevant events. No remote administrative commands exist to modify or delete log files; only the local root admin is authorized to interact with these files. A Security Administrator is therefore authorized to manually delete audit records only if they are authenticated to the TOE via the local console. Each of the log files are rotated such that the exhaustion of storage for one file causes it to be archived with active auditing moving to the next file in the rotation. Once all files are filled in this manner, subsequent rotations will overwrite the oldest stored record. Each of the log files have a maximum file size of 10 MB and store five separate files in the rotation. In the evaluated configuration, all log files are configured to transmit log data to a remote syslog server over TLS; this occurs in real-time, simultaneously with the audit records that are written locally. In the event of a syslog outage, there is no buffering mechanism that allows for synchronization between local and remote logs.

6.2 Cryptographic Support

The TOE implements cryptographic functionality using VMware's OpenSSL FIPS Object Module 2.0.20-vmw and VMware's BC-FJA (Bouncy Castle FIPS Java API) 1.0.2.3 on JDK 11. The TOE uses OpenSSL for outbound TLS connections to environmental syslog servers and key pair generation for certificate signing requests. The TOE also uses OpenSSL for outbound Blast media communications with environmental Horizon Agents over HTTPS. The TOE uses Bouncy Castle BC-FJA for the initial HTTPS connection establishment and Blast media communications for an inbound Horizon Client connection, outbound HTTPS Connection Server communications, and the admin web server.

Table 6 below identifies the algorithm certificates that apply to all cryptographic libraries.

Table 6: Validated Algorithm Implementations

Functions	Libraries	Standards	Certificates
Asymmetric key generation (FCS_CKM.1)			
RSA Schemes (3072-bit)	OpenSSL	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	A1292 (OpenSSL)
ECC Schemes (ECDSA P-256, P-384, P-521 curves)	OpenSSL BC-FJA	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	A1292 (OpenSSL) A2841 (BC-FJA)

Functions	Libraries	Standards	Certificates
FFC Schemes ('safe-prime' groups)	BC-FJA	NIST SP 800-56A Revision 3; RFC 7919	A2841 (BC-FJA)
Key Establishment (FCS_CKM.2)			
Elliptic curve-based scheme	OpenSSL BC-FJA	NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography	A1292 (OpenSSL) A2841 (BC-FJA)
Finite field-based scheme	BC-FJA	NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography; RFC 7919	A2841 (BC-FJA)
Encryption/Decryption (FCS_COP.1/DataEncryption)			
AES in CBC mode (128, 256 bits)	OpenSSL BC-FJA	AES as specified in ISO 18033-3 CBC as specified in ISO 10116	A1292 (OpenSSL) A2841 (BC-FJA)
AES in GCM mode (128, 256 bits)	OpenSSL BC-FJA	AES as specified in ISO 18033-3 GCM as specified in ISO 19772	A1292 (OpenSSL) A2841 (BC-FJA)
Cryptographic signature services (Signature Generation and Verification) (FCS_COP.1/SigGen)			
RSA Digital Signature Algorithm (rDSA) (3072-bit modulus)	OpenSSL BC-FJA	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,	A1292 (OpenSSL) A2841 (BC-FJA)
Cryptographic hashing (FCS_COP.1/Hash)			
SHA-1 (digest size 160 bits)	OpenSSL	ISO/IEC 10118-3:2004	A1292 (OpenSSL)
SHA-256 (digest size 256 bits) SHA-384 (digest size 384 bits) SHA-512 (digest size 512 bits)	OpenSSL BC-FJA	ISO/IEC 10118-3:2004	A1292 (OpenSSL) A2841 (BC-FJA)
Keyed-hash message authentication (FCS_COP.1/KeyedHash)			
HMAC-SHA-1 (key size 512 bits, digest size 160 bits)	OpenSSL	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2	A1292 (OpenSSL)

Functions	Libraries	Standards	Certificates
HMAC-SHA-256 (key size 512 bits, digest size 256 bits) HMAC-SHA-384 (key size 512 bits, digest size 384 bits)	OpenSSL BC-FJA	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2	A1292 (OpenSSL) A2841 (BC-FJA)
Random bit generation (FCS_RBG_EXT.1)			
CTR-DRBG (AES) – 256 bits entropy	OpenSSL	ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions	A1292 (OpenSSL)
Hash_DRBG(SHA-512) – 256 bits entropy	BC-FJA	ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions	A2841 (BC-FJA)

The TOE generates asymmetric keys for TLS and X.509 certificates. The TOE generates ECC keys using P-256, P-384, and P-521 and FFC keys using ffdhe groups 2048, 3072, 4096, 6144, and 8192 in support of TLS key establishment as a client. As a server, the TOE uses P-384 for ECC keys. The TOE generates 3072-bit RSA keys as part of generating certificate signing requests per FIA_X509_EXT.3. The TOE performs cryptographic key establishment for TLS using elliptic curves or safe primes in a manner that conforms to NIST SP 800-56A revision 3. All services where the TOE acts as a TLS/HTTPS server (FCS_TLSS_EXT.1, FCS_TLSS_EXT.2) use ECDH key establishment. All services where the TOE acts as a TLS or TLS/HTTPS client (FCS_TLSC_EXT.1) can use ECDH key establishment. Services that use BC-FJA (i.e. communications with Horizon Connection Server) can also use DH key establishment, depending on the negotiated cipher suite. The TOE also implements the cryptographic algorithms listed below in support of TLS, each of which are implemented by both of the cryptographic libraries that the TOE uses. Note that TLS client functionality supports all algorithms listed below, while TLS server functionality only supports a subset of those algorithms where noted.

- AES-CBC, AES-GCM (128-bit, 256-bit)
    - TLS server functionality uses 256-bit AES-GCM only
- RSA signature generation and verification (3072 bit)
- SHA-256, SHA-384
    - TLS server functionality uses SHA-384 only
- HMAC-SHA-256 (512 bit key and block size, 256 bit output length), HMAC-SHA-384 (512 bit key and block size, 384 bit output length)
    - TLS server functionality uses HMAC-SHA-384 only

The TOE's implementation of OpenSSL uses CTR_DRBG(AES) and the TOE's implementation of BC-FJA uses Hash_DRBG(SHA-512).

Outside of trusted communications, the TOE also uses the following cryptographic algorithms:

- SHA-384: hashing of remote admin password data
- SHA-512: hashing of local admin password data
- HMAC-SHA-1 with SHA-1: OpenSSL software integrity verification (512 bit key and block size, 160 bit output length)

- HMAC-SHA-256 with SHA-256: BC-FJA software integrity verification (512 bit key and block size, 256 bit output length)

The TOE includes both plaintext static keys that reside in nonvolatile memory and plaintext ephemeral keys that reside in volatile memory. Keys stored in volatile memory include the TLS pre-master secret, session key, and session authentication key. When these are no longer needed, they are destroyed through destruction of reference to the key followed by a request for garbage collection. OpenSSL does this using API function calls and BC-FJA does this by invoking the JVM garbage collector on thread termination; there are no circumstances where the claimed behavior does not apply.

The only static key maintained by the TOE is the private key portion of the TOE's TLS server certificate. This key originates when it is generated by the TSF as part of generating a certificate signing request per FIA_X509_EXT.3. When the request is generated, the private key is temporarily stored in plaintext. Once the signed certificate response is received, the certificate and private key are loaded into the password-protected Java Keystore (see section 6.5). The BC-FJA FIPS keystore uses AES-256, SHA-512, and PBKDF2 to ensure that the stored data is inaccessible without a valid password to unlock it. The private key is then destroyed through the TOE's invocation of OS mechanisms to destroy the file system object that represents the key; there are no circumstances where this claimed behavior does not apply.

The TOE implements the ISO/IEC 18031:2011 Deterministic Random Bit Generator (DRBG) based on the AES 256 block cipher in counter mode (CTR_DRBG (AES-256)) for OpenSSL and a SHA-512 Hash_DRBG for Bouncy Castle BC-FJA. The TOE instantiates the DRBG with maximum security strength, obtaining a minimum of 256 bits of entropy drawn from /dev/random to seed the DRBG. The TOE obtains platform entropy data through the processor RDRAND instruction set, described in the proprietary Entropy Design document. The VMware ESXi hypervisor in the Operational Environment provides a passthrough interface to underlying platform hardware. The TOE uses its DRBGs to generate all keys.

The TOE Implements several protocols for external communications, as described below.

The TOE implements TLS 1.2 as both a client and a server, rejecting all other TLS/SSL versions. Specifically, the use of the "FIPS" build as the evaluated configuration of the TOE (see section 2.4.1) forces the TOE to use TLS 1.2; configuration options are present for TLS 1.0, 1.1, and 1.3, but they are permanently greyed out in this build. Pre-TLS SSL versions are automatically disabled by default and cannot be enabled under any circumstance.

The TOE implements different external interfaces using OpenSSL and BC-FJA. Section 6.7 identifies each external interface, including the cryptographic library used to implement the trusted protocol for that interface. In the evaluated configuration, the supported ciphersuites are configured to include the following:

- When acting as TLS client:
  - For all interfaces (OpenSSL and BC-FJA):
    - 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
  - For BC-FJA interfaces only:
    - TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
    - TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246

        - TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
        - TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- When acting as TLS server:
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

In the evaluated configuration, the TOE is configured to support the following parameters for TLS. The TLS server uses secp384r1 as the curve used for key establishment. The TLS client can use any of secp256r1, secp384r1, or secp521r1 when a TLS_ECDHE cipher suite is negotiated. For interfaces that use BC-FJA, the client also supports the RFC7919 'safe-prime' key generation methods ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, and ffdhe8192 when a TLS_DHE cipher suite is negotiated.

Regardless of the cryptographic library used to implement TLS client functionality, when acting as a client, the TOE does not support mutual authentication, so it does not present a TLS client certificate. The TOE validates any presented server certificate in the manner specified by RFC 6125 section 6. Specifically, the client uses the FQDN (host name) in the CN or SAN as the reference identifier for the TLS server certificate. IP addresses and wildcards are not supported. Invalid certificates are rejected. No administrator override exists to re-adjudicate the rejection of an invalid certificate.

When acting as a TLS server, the TOE supports mutual authentication for inbound connectivity from remote Horizon Clients. The reference identifier is based on the CN of the certificate and the entire public key portion of the certificate. Specifically, a hash is computed from the public key portion of the certificate and associated with the UPN or email address that is contained in the CN field of the client certificate. Subsequent connection attempts with that CN must use the same certificate. If the presented client certificate is invalid for any reason, the connection is rejected. No administrator override exists to re-adjudicate the rejection of an invalid certificate. The TOE does not support session resumption based on either session IDs or session tickets for any TLS interfaces.

The TOE also implements HTTPS for inbound connectivity from remote administrators and Horizon Clients, and for outbound connectivity to Horizon Connection Servers. All HTTPS implementation is compliant with RFC 2818 and uses the TLS functionality described above. In all cases, an invalid certificate will cause the connection to be aborted.

6.3 Identification and Authentication

The TOE has two management interfaces: a remote web GUI and a local-only console, which is accessed via VMware vCenter. The remote web GUI locks out administrators after several failed authentication attempts; the threshold for this is a configurable value between 1 and 100, with a default value of 3. The TSF increments the failure counter in non-volatile memory so that the accumulated number of failures is persisted across reboots. In the event the remote administrator is locked out, access is restored by a local admin issuing the 'supervisorctl restart admin' command or by waiting for the lockout period to expire. The lockout period is a configurable value between 1 and 9999 minutes, with a default value of 5. The local administrator account is not subject to lockout.

The remote web GUI and local console use two different sets of credentials ("admin" for remote administration, "root" for local console), but both are local password-based authentication mechanisms. Password characters are obfuscated on entry for both interfaces. The allowed character sets for each are uppercase letters, lowercase letters, numbers, and the special characters !, @, #, $, %, ^, &, *, (, and

). In the evaluated configuration, the minimum password length for each is configurable between 8 and 64 characters, and the maximum password length is 64 characters. For both local and remote management interfaces, there is no TSF-mediated function that will be allowed prior to authentication aside from display of the warning banner to the administrator. In both cases, a successful logon is measured by the computing of the hash of the supplied password and comparing it to the expected result. Remote user authentication occurs at the application layer and is unrelated to the establishment of the underlying trusted path.

The TOE performs X.509 certificate validation in support of TLS and TLS/HTTPS communications. As both the OpenSSL and Bouncy Castle BC-FJA libraries implement TLS, both libraries implement the same X.509 validation functions. All certificates that are presented to the TOE are validated. This includes TLS server certificates for all cases where the TOE acts as a TLS client, TLS client certificates when inbound Horizon Client connections are made, and the TOE's own TLS server certificate when a CA response to a certificate signing request is provided back to the TOE. Specifically, the following validation rules are followed:

- The TSF performs RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The TSF validates that the certification path terminates with a trusted CA certificate designated as a trust anchor.
- The certification path is validated by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The extendedKeyUsage field is validated according to the following rules:
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

Note that FIA_X509_EXT.1/Rev also optionally requires the TSF to verify the extendedKeyUsage field for certificates used for trusted updates and executable code integrity verification. The TOE does not use X.509 certificates for these purposes so this portion of the requirement is trivially satisfied by the TSF.

The TOE chooses the certificate to present to external TLS clients based on the server certificate that is loaded into it as part of administrative configuration. In all other cases the TOE validates the certificate that is presented to it and validates the chain based on what is included in the certificate.

The TOE supports the use of CRL for revocation status checking of TLS certificates. In all cases, if the revocation status of a certificate cannot be determined, the TSF treats it is invalid. Both the leaf certificate and any intermediate CA certificates are checked for TLS.

The TOE also includes the ability to generate a certificate signing request as specified by RFC 2986 to be sent to a CA for issuance of a TLS server certificate. The TOE uses OpenSSL as the mechanism that supports this. The certificate signing request generates an RSA key pair and can include the Common Name, Organization, Organizational Unit, and Country fields. When the signed response is loaded into the TOE, the TSF validates the certificate chain from the root CA and will accept the response as the TLS server certificate if the certificate chain is valid. This is subsequently the certificate that the TOE issues to external TLS clients attempting to connect to it.

6.4 Security Management

The TOE provides management functionality over both local and remote interfaces. The local interface is a direct console interface to the TOE's OS platform and the remote interface is a web GUI accessed over TLS/HTTPS. The local and remote interfaces have different roles with different credentials; the local account is an OS root user and the remote account is specifically for the UAG management application itself. Both accounts function as Security Administrators for the management functions that are available on their respective interfaces. There are no lesser-privileged management roles on the TOE; all administrative access to the TOE is by Security Administrators. Not all functions may be performed on both interfaces; Table 7 below identifies the management functions that are available on each interface.

Table 7: Management Functions by Interface

Function	Local	Remote
Ability to administer the TOE locally	X	
Ability to administer the TOE remotely		X
Ability to configure the access banner	X	X
Ability to configure the session inactivity time before session termination		X
Ability to update the TOE and verify the updates using hash comparison prior to installing those updates	X	X
Ability to configure authentication failure parameters for FIA_AFL.1		X
Ability to modify the behavior of the transmission of audit data to an external IT entity		X
Ability to manage the cryptographic keys	X	X
Ability to re-enable an Administrator account	X	
Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors		X
Ability to import X.509v3 certificates to the TOE's trust store		X

The Security Administrator is able to configure the TOE's interactions with external entities. Specifically, they are able to determine and modify the behavior of the TOE's connectivity to an external syslog server. The Security Administrator also has the ability to manage cryptographic keys by using the remote web GUI to manipulate the certificates that reside in the TOE's trust store and by using the local console to generate certificate signing requests, which include key pairs.

6.5 Protection of the TSF

The TOE implements various self-protection mechanisms for its functions and data.

The TSF stores all password data in an obfuscated format. Specifically, remote administrator password data is stored using PBKDF2 password-based encryption with SHA-384, and local administrator password data is stored using the Linux pam.d module as a SHA-512 hash. The TSF stores its private key on its local file system when the key pair for the CSR is first generated. When the signed certificate response is received by the TOE, the certificate and private key are loaded into the password-protected Java

Keystore. For both credential and non-public key data, no administrative interface exists to read this data in plaintext form.

As a virtual network device, the TOE obtains its time data from the virtualization system on which it is deployed using the VMware Tools periodic time synchronization interface. This interface is invoked once per minute to determine if there is a mismatch in system time between the host and the VM and either updates the VM's clock immediately (if behind the host clock) or slows down the VM's clock to match the host (if ahead of the host clock). A sync operation happens once every minute regardless of drift; there is no minimum threshold of drift required to trigger a correction and the maximum possible drift is one minute. The TSF uses time data for audit log timestamps, timed administrator lockouts, idle session timeouts, and validation of X.509 certificates.

To verify its own integrity, the TOE performs a file system integrity test on the boot partition using fsck and cryptographic module self-tests during initial start-up for both OpenSSL and Bouncy Castle BC-FJA. The file system integrity check ensures that the contents of the boot partition are not corrupted. The cryptographic module self-tests that are performed at start-up are as follows:

- Software integrity test – used to verify that the cryptographic code has not been modified
  - HMAC-SHA-256 for BC-FJA
  - HMAC-SHA-1 for OpenSSL
- Cryptographic known answer tests – used to verify that cryptographic operations yield expected results with known inputs
  - AES encryption/decryption
  - RSA signature generation/verification
  - DRBG known answer test
  - ECC known answer test (per FIPS 140-2 IG 9.6)

In addition to this, both cryptographic modules perform the following conditional self-tests:

- RNG tests – used to ensure sequences of pseudorandom data are not predictable
  - Continuous RNG test for stuck fault
  - DRBG health testing as required by NIST SP 800-90A section 11
- Pairwise consistency tests – used to ensure that for a given public/private key pair, a piece of data encrypted with a public key and subsequently decrypted with a private key is the same before encryption and after decryption
  - Performed on each RSA and ECDSA key pair generation

These tests are sufficient to ensure that the TSF executable code have appropriately not been modified, and that the cryptographic functionality has not been tampered with or degraded, either through a compromise of the cryptographic functionality itself or through a degradation of any hardware components on which this functionality relies. There is no situation in which an administrator may unwittingly be using a modified TOE or may be using the TOE to transmit sensitive data using a degraded cryptographic implementation.

When a self-test fails, information will be communicated in the following manner:

- BC-FJA failures logged under the "org.bouncycastle.crypto.fips.FipsSelfTestFailedError" message type in various log files in the /opt/vmware/gateway/logs directory.

- OpenSSL failures logged into the syslog-ng.log or bsg.log in the /opt/vmware/gateway/logs directory.
- Failure of boot partition filesystem check will cause UAG to fail to boot entirely and a corresponding error message will be displayed on the local console.

Any cryptographic failures will also result in the inability to establish trusted communications so the need to check logs for troubleshooting purposes will be evident from this.

The TOE supports software/firmware updates. Only one version of the TOE software/firmware is loaded at any one time. The overall TOE version (e.g. 2209.2) can be checked through the remote administrative GUI, and the versions of individual packages can be checked through the 'tdnf list intalled' command on the local console. The TOE has a manual update mechanism. Package updates to apply security fixes for incremental updates are obtained by the administrator from packages.vmware.com along with the corresponding updates-fips.json file. These packages are hashed using SHA-256; their hash values are checked manually by the administrator. If found to be valid, the administrator places the packages and the updates-fips.json file on a file server that the TOE is configured to check for available updates. The Security Administrator then uses the web GUI to set the TOE to update on next boot and then reboots the TOE. Successful and failed updates are captured in the package-updates.log file.

6.6 TOE Access

The TOE enforces access restrictions on both its local and administrative interfaces. The local console and admin web GUI both have configurable idle session timeout periods – from 30-3600 seconds for the local console (default 300 seconds) and from 1-1440 minutes for the Admin UI (default 10 minutes). When the idle session timeout period has elapsed, the session is terminated. Each interface also includes mechanisms for the Security Administrator on each to voluntarily terminate their own session (logout button on GUI, 'exit' command on CLI). The idle session timeout value is independently configurable on each interface. Both local and remote interfaces additionally display a configurable pre-authentication warning banner that can be used to advise Security Administrators of appropriate usage of the system. Each interface has its own independently configurable banner message.

6.7 Trusted Path/Channels

As stated in section 6.2, the TOE implements both OpenSSL 2.0.20-vmw and Bouncy Castle BC-FJA 1.0.2.3 as cryptographic libraries in support of trusted communications. The following are the trusted paths and channels implemented by the TOE along with the cryptographic library the TSF uses to implement them:

- Trusted channels (FTP_ITC.1)
  - Remote syslog server
    - TLS (ephemeral TCP port determined by server)
    - No client authentication
    - TOE is client
    - Implemented by OpenSSL
  - Horizon Client user authentication (XML API channel)
    - TLS/HTTPS (TCP 443)
    - Client authentication

        - TOE is server
        - Implemented by Bouncy Castle BC-FJA
    - Horizon Client media transmission (Blast channel)
        - TLS/HTTPS (TCP 443)
        - Client authentication
        - TOE is server
        - Implemented by Bouncy Castle BC-FJA
    - Horizon Connection Server session establishment
        - TLS/HTTPS (ephemeral TCP port determined by server)
        - No client authentication
        - TOE is client
        - Implemented by Bouncy Castle BC-FJA
    - Horizon Agent media transmission (Blast channel)
        - TLS/HTTPS (ephemeral TCP port determined by server)
        - No client authentication
        - TOE is client
        - Implemented by OpenSSL
- Trusted paths (FTP_TRP.1/Admin)
    - Remote administration
        - TLS/HTTPS (TCP/9443)
        - No client authentication
        - TOE is server
        - Implemented by Bouncy Castle BC-FJA

The TSF initiates trusted channel functions for transmission of audit data to external storage and to communicate with an environmental Horizon Connection Server and Horizon Agents on behalf of environmental Horizon Clients.

For non-administrative usage of the TOE, the TOE implements trusted channels to facilitate connectivity between a Horizon Client and a Horizon Agent. The general workflow for this is as follows:

1. Horizon Client connects to UAG via XML API channel.
2. UAG authenticates Horizon Client user via mutual TLS authentication and passes access request to Connection Server via SAML assertion.
3. Connection Server communicates with connected Horizon Agent(s) to determine which Agent(s) should be used to serve the requested media to the Horizon Client.
4. Connection Server provides UAG with single-use authorization token to access authorized Horizon Agent(s), which is passed back to Horizon Client over via XML API channel.
5. Horizon Client opens TLS connections to UAG for one or more media channels over Blast based on the desired media services.
6. UAG validates the Horizon Client’s authorization token and passes to the designated Horizon Agent(s).
7. Horizon Agent(s) validates the provided authorization token and allows for establishment of the TLS session(s) for the desired channels.
8. The UAG maintains both the ‘internal’ and ‘external’ channels to allow the Horizon Client to interact with the Horizon Agent(s).

With respect to the TOE boundary, the primary consideration is that all channels into and out of the TOE for the establishment of this connectivity use TLS or TLS/HTTPS; the workflow is provided to show the circumstances in which each channel is used.

7 Protection Profile Claims

This ST is conformant to the *collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020* (NDcPP).

The TOE consists of a virtual network device, which includes the operating system of the device within its boundary.

As explained in section 3, Security Problem Definition, the Security Problem Definition of the NDcPP has been included by reference into this ST.

As explained in section 4, Security Objectives, the Security Objectives of the NDcPP has been included by reference into this ST.

All claimed SFRs are defined in the NDcPP. All mandatory SFRs are claimed. Selection-based SFR claims are consistent with the selections made in the mandatory SFRs that prompt their inclusion. Some optional SFR claims are made at the TOE developer’s discretion.

8 Rationale

This Security Target includes by reference the NDcPP Security Problem Definition, Security Objectives, and Security Assurance Requirements. The Security Target does not add, remove, or modify any of these items. Security Functional Requirements have been reproduced with the Protection Profile operations completed. All selections, assignments, and refinements made on the claimed Security Functional Requirements have been performed in a manner that is consistent with what is permitted by the NDcPP. The proper set of selection-based requirements have been claimed based on the selections made in the mandatory requirements. Consequently, the claims made by this Security Target are sufficient to address the TOE's security problem. Rationale for the sufficiency of the TOE Summary Specification is provided below.

8.1 TOE Summary Specification Rationale

This section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions meet the TOE security requirements. Each description includes rationale indicating which requirements the corresponding security functions satisfy. The combined security functions work together to satisfy all of the security requirements. The security functions described in Section 6 are necessary for the TSF to enforce the required security functionality. Table 8 demonstrates the relationship between security requirements and functions.

Table 8: Security Functions vs. Requirements Mapping

	Security Audit	Cryptographic Support	Identification and Authentication	Security Management	Protection of the TSF	TOE Access	Trusted Path/Channels
FAU_GEN.1	X						
FAU_GEN.2	X						
FAU_STG.1	X						
FAU_STG_EXT.1	X						
FCS_CKM.1		X					
FCS_CKM.2		X					
FCS_CKM.4		X					
FCS_COP.1/DataEncryption		X					
FCS_COP.1/Hash		X					
FCS_COP.1/KeyedHash		X					
FCS_COP.1/SigGen		X					
FCS_HTTPS_EXT.1		X					
FCS_RBG_EXT.1		X					
FCS_TLSC_EXT.1		X					
FCS_TLSS_EXT.1		X					

	Security Audit	Cryptographic Support	Identification and Authentication	Security Management	Protection of the TSF	TOE Access	Trusted Path/Channels
FCS_TLSS_EXT.2		X					
FIA_AFL.1			X				
FIA_PMG_EXT.1			X				
FIA_X509_EXT.1/Rev			X				
FIA_X509_EXT.2			X				
FIA_X509_EXT.3			X				
FIA_UIA_EXT.1			X				
FIA_UAU.7			X				
FIA_UAU_EXT.2			X				
FMT_MOF.1/Functions				X			
FMT_MOF.1/ManualUpdate				X			
FMT_MTD.1/CoreData				X			
FMT_MTD.1/CryptoKeys				X			
FMT_SMF.1				X			
FMT_SMR.2				X			
FPT_APW_EXT.1					X		
FPT_SKP_EXT.1					X		
FPT_STM_EXT.1					X		
FPT_TST_EXT.1					X		
FPT_TUD_EXT.1					X		
FTA_SSL.3						X	
FTA_SSL.4						X	
FTA_SSL_EXT.1						X	
FTA_TAB.1						X	
FTP_ITC.1							X
FTP_TRP.1/Admin							X