VirtualWisdom Platform Appliance v5.7 Security Target

Virtual Instruments
2331 Zanker Road
San Jose, CA 95131
408.579.4000
www.virtualinstruments.com

Copyright © 2019 by Virtual Instruments Corporation

Virtual Instruments Corporation
2331 Zanker Road
San Jose, CA 95131

Phone: 1-408-579-4000
Fax: 1-408-579-4001

Business Hours: Monday - Friday, 8:00 am- 5:00 pm Pacific Time

Trademarks

Document Revisions

Rev #	Date	Software	Description
0.1	10/20/2017	5.6	Initial draft of this document.
0.2	03/02/2018	5.6	Updated security assumptions and objectives. Corrected SFR names. Added information to the crypto claims in TSS section. Updated rationale section for the applicable technical decisions. Added Appendix A.
0.3	03/07/2018	5.6	Minor text changes.
0.4	03/08/2018	5.6	Minor text additions.
0.5	06/01/2018	5.7	Changed software version to 5.7. Removed the list of excluded functionalities. Updated text for NDcPP v2.0 Errata. Added new technical decisions. Updated formatting.
0.6	06/28/2018	5.7	Updated TOE reference. Added secure NTP server to security claims.
0.7	09/06/2018	5.7	Updated product naming. Removed TLS cipher suites from the claims. Added CAVP certificate references. Added new technical decisions. Minor text additions.
0.8	10/19/2018	5.7	Updated the look and layout of the document. Numbered references were updated to word references.
0.9	11/15/2018	5.7	Updated Figure 2 and claims.
1.0	12/06/2018	5.7	Added numbering to section titles and made minor change to sections 8.1.2.1 and 8.1.3 of the TSS.
1.1	02/25/2019	5.7	Updated Table 11.

Contents

Chapter 1

Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the organization of the ST. The TOE is the Virtual Instruments VirtualWisdom Platform Appliance and will hereafter be referred to as the TOE. The TOE is an infrastructure performance management (IPM) appliance.

1.1. Purpose

This ST is divided into ten chapters, as follows:

- Introduction (Chapter 1) – Provides a brief summary of the ST contents and describes the organization of other sections within this document. It also provides an overview of the TOE security functionality and describes the physical and logical scope for the TOE as well as the ST and TOE references.
- Conformance Claims (Chapter 2) – Provides the identification of any Common Criteria (CC), Protection Profile (PP), and Evaluation Assurance Level (EAL) package claims. It also identifies whether the ST contains extended security requirements.
- Security Problem Definition (Chapter 3) – Describes the threats, Organizational Security Policies (OSPs), and assumptions that pertain to the TOE and its environment.
- Security Objectives (Chapter 4) – Identifies the security objectives that are satisfied by the TOE and its environment.
- Extended Components (Chapter 5) – Identifies new components (extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs)) that are not included in CC Part 2 or CC Part 3.
- Security Assurance Requirements (Chapter 6) – Presents the SARs met by the TOE.
- Security Functional Requirements (Chapter 7) – Presents the SFRs met by the TOE.
- TOE Summary Specification (Chapter 8) – Describes the security functions provided by the TOE that satisfy the SFRs and objectives.
- Rationale (Chapter 9) - Presents the rationale for the SFR dependencies as to their consistency, completeness, and suitability.
- Acronyms and Terms (Chapter 10) – Defines the acronyms and terminology used within this ST.

1.2. Security Target and TOE References

Table 1 below shows the ST and TOE references.

Table 1 – ST and TOE References

ST Title	Virtual Instruments VirtualWisdom Platform Appliance v5.7 Security Target
ST Version	Version 1.0
ST Author	Corsec Security, Inc.
ST Publication Date	12/6/2018
TOE Reference	Virtual Instruments VirtualWisdom Platform Appliance v5.7

1.3. Product Overview

The Product Overview provides a high-level description of the Virtual Instruments VirtualWisdom Platform Appliance that is the subject of the evaluation. The following section, TOE Overview, will provide the introduction to the parts of the overall product offering that are specifically being evaluated.

The VirtualWisdom Platform Appliance is an IPM appliance that provides real-time and historical insights into the performance, availability, health, and utilization of a customer's data center infrastructure. It collects, correlates, and analyzes real-time data from VM[1]s, FC[2] switches, and SAN[3] and NAS[4] networks, to provide a complete and accurate view of the end-to-end system. This allows customers to optimize their data center infrastructure and proactively identify and resolve issues. VirtualWisdom Platform Appliance's Applied Analytics feature also analyzes the collected data to optimally balance host workload and traffic, investigate events that violate preconfigured metrics, and provide recommendations to optimize system-wide performance.

The VirtualWisdom Platform Appliance v5.7 firmware is preinstalled on the appliance and provides the web-based VirtualWisdom UI[5] for management. The VirtualWisdom Platform Appliance v5.7 firmware also includes the Virtual Server, Network Switch, and NTAP[6] Storage software probes that are used to collect data from VMs, FC switches, and NAS storage arrays, respectively. The firmware can also be integrated with Virtual Instruments' own hardware-based SAN and NAS Performance Probes to collect data from SAN and NAS networks.

The VirtualWisdom Platform Appliance includes a management port used for initial configuration and five NIC[7] port used for all VirtualWisdom network traffic, including administration traffic, audit traffic, and communications with SAN and NAS Performance Probes. Figure 1 represents the VirtualWisdom Platform Appliance 4220.

Figure 1 – VirtualWisdom Platform Appliance 4220

1.4. TOE Overview

The TOE Overview summarizes the usage and major security features of the TOE. The TOE Overview provides a context for the TOE evaluation by identifying the TOE type, describing the product, and defining the specific evaluated configuration.

The TOE is the VirtualWisdom Platform Appliance, a network device consisting of hardware and software that provides IPM. Its security features include securing remote management, providing identification and authentication services for both local and remote logins, auditing security-related events, cryptographically validating the source of any update, and offering protection against common network-based attacks.

The TOE employs TLS[8] v1.2 to protect the communication path to external entities and X.509 certificates for authentication of secure channels. The TOE is remotely managed in a secure manner via the VirtualWisdom Platform Appliance UI.

[1] VM – Virtual Machine
[2] FC – Fibre Channel
[3] SAN – Storage Area Network
[4] NAS – Network-Attached Storage
[5] UI – User Interface
[6] The NTAP software probe collects data from the NetApp NAS arrays.
[7] NIC – Network Interface Card
[8] TLS – Transport Layer Security

1.4.1. TOE Environment

The TOE relies on non-TOE hardware and software for its essential operation. Though this hardware and software is necessary for the TOE's operation, it is not part of the TOE. The following non-TOE hardware and software is required for essential operation of the TOE:

- DNS[9] server
- NTP[10] server
- Syslog server
- Local management workstation with Ethernet port and Google Chrome 21+, Firefox 15+, Internet Explorer 10+ or Safari 6+ web browser
- Remote Windows 7 (or higher) 64-bit management workstation with Google Chrome 21+, Firefox 15+, Internet Explorer 10+ or Safari 6+ web browser
- 1 straight Ethernet cable for a direct connection to the management port from the local management workstation
- 1 straight Ethernet cable for connecting the internal network with the remote management workstation to the TOE (via the appliance's NIC0 port)

1.5. TOE Description

This section primarily addresses the physical and logical components of the TOE that are included in the evaluation.

1.5.1. Physical Scope

Figure 2 illustrates the physical scope and the physical boundary of the overall solution and ties together all of the components of the TOE and the constituents of the TOE environment. The TOE is a hardware and firmware TOE; its components are the VirtualWisdom Platform Appliance and the VirtualWisdom v5.7 firmware (preloaded on the appliance). The VirtualWisdom Platform Appliance 4220 is the appliance model included in the evaluated configuration. The following previously undefined acronym appears in Figure 2:

- HTTPS – Hypertext Transfer Protocol Secure

[9] DNS – Domain Name System

[10] NTP – Network Time Protocol

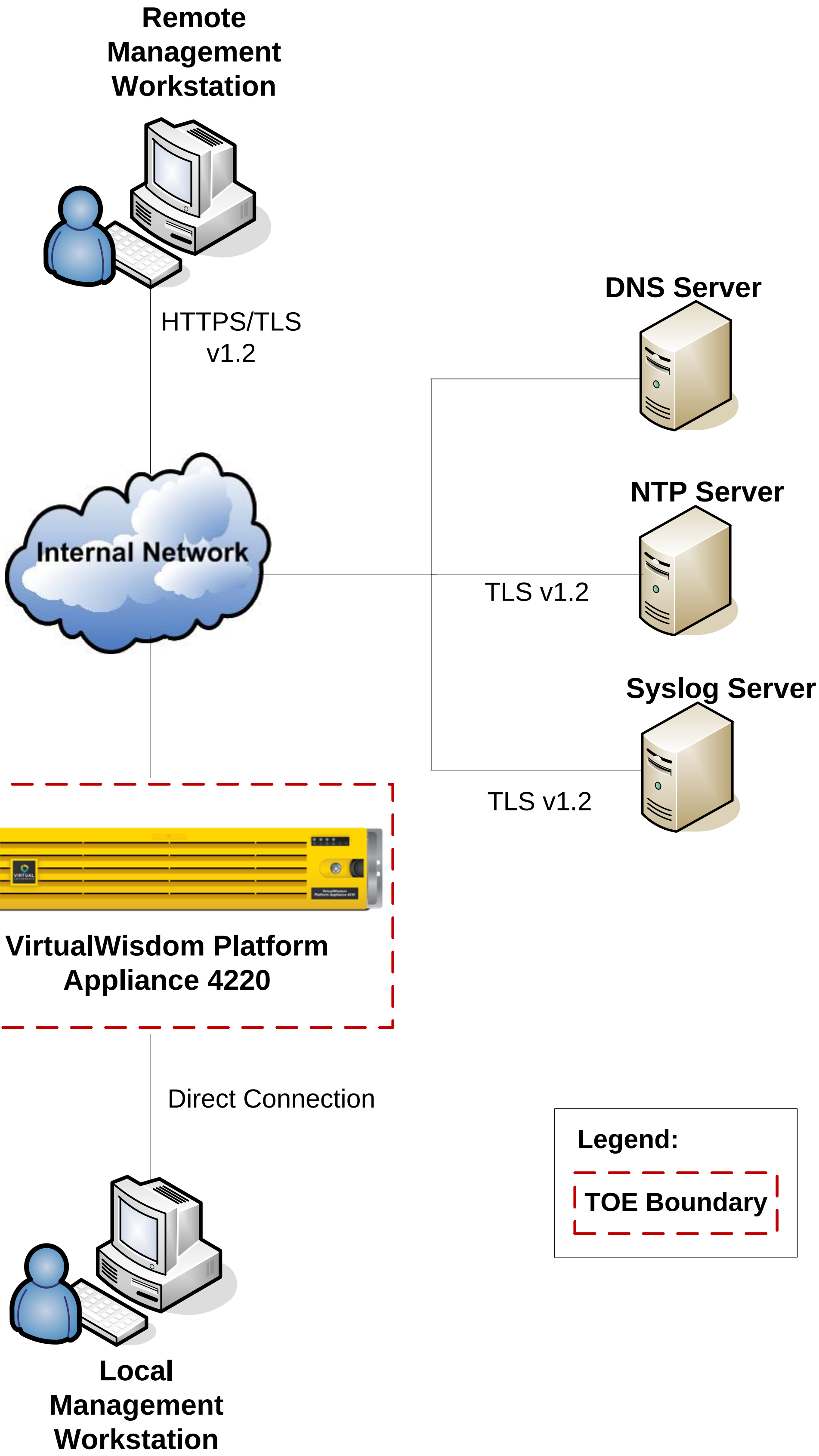

Figure 2 – Physical TOE Boundary

The TOE boundary includes all the VirtualWisdom Platform Appliance parts that were developed by Virtual Instruments. Any third-party source code or firmware on the VirtualWisdom Platform Appliance that Virtual Instruments has modified is considered to be TOE firmware. The TOE Boundary specifically does not include any of the third-party software that the TOE relies upon as described in TOE Environment section of the ST.

1.5.1.1. TOE Hardware and Firmware

The TOE is a hardware and firmware TOE with the following components:

- VirtualWisdom Platform Appliance Firmware Image v5.7
- VirtualWisdom Platform Appliance 4220

For the evaluated configuration, the TOE firmware is pre-installed and runs on the Virtual Instruments VirtualWisdom Platform Appliance 4220. The Virtual Instruments VirtualWisdom Platform Appliance 4220 is shipped to customers via courier delivery. The VirtualWisdom Platform Appliance Firmware v5.7, pre-installed on the appliance, may also be acquired by contacting Virtual Instruments Technical Support.

1.5.1.2. Guidance Documentation

Table 2 lists the TOE Guidance Documentation needed to install, configure, and maintain the TOE. Customers may obtain the TOE Guidance Documentation by contacting Virtual Instruments Customer Support. It is provided in PDF format.

Table 2 – Guidance Documentation

Document Name	Description
Virtual Instruments VirtualWisdom Platform Appliance Platform Appliance 4220 Installation Guide	Includes steps for the basic initialization and setup of the TOE.
Virtual Instruments VirtualWisdom Platform Appliance 5.7 User Guide	Contains detailed steps for how to properly configure and maintain the TOE.
Virtual Instruments VirtualWisdom Platform Appliance v5.7 Guidance Documentation Supplement v0.5	Contains information regarding the specific configuration for the TOE evaluated configuration.

1.5.2. Logical Scope

The logical boundary of the TOE will be broken down into the following security classes, which are further described in chapters 7 and 8 of this ST.

1.5.2.1. Security Audit

The TOE generates audit records for security relevant actions of the authorized administrators within the VirtualWisdom Platform Appliance UI. Audit records are sent in real-time to an external Syslog server over a secure channel. The audit records include the date and time of the events, type of events, subject identity, and outcome of the events. When the local storage space approaches its size limit, the earliest audit records are overwritten. Local audit records can be downloaded to be viewed by authorized administrators and are protected from unauthorized modification or deletion.

1.5.2.2. Cryptographic Support

The Cryptographic Support of the TSF[11] function provides cryptographic functions to secure TLS v1.2 and HTTPS connections from external hosts connecting to the TOE via the VirtualWisdom Platform Appliance UI. Cryptographic functions are also used to secure TLS v1.2 trusted channels between the TOE and the Syslog and NTP servers.

The TOE supports AES[12] 128-bit and 256-bit CBC[13] and GCM[14] modes for encryption and decryption. The TOE supports RSA[15] and ECDSA[16] signature generation and verification.

The TOE supports the generation of asymmetric cryptographic keys and uses ECDHE[17] for key establishment. The TOE uses a CTR[18]-based DRBG[19] and destroys plaintext keys in both volatile and non-volatile storage.

1.5.2.3. Identification and Authentication

The TOE provides functionality that requires administrators to verify their claimed identity. The Identification and Authentication TSF ensures that only legitimate administrators can gain access to the configuration and management settings of the TOE. Besides loading the VirtualWisdom Platform Appliance Login page and interacting with the login banner, administrators must log in with a valid user name and password before management of the TOE is permitted.

The TOE uses X.509 certificates for TLS v1.2 communications. These certificates are validated by the TOE and used to support authentication for TLS v1.2. The TOE supports the generation of Certificate Request Messages (CRMs).

The TOE implements requirements for password complexity and length and provides obscured feedback to administrative users while local VirtualWisdom Platform Appliance UI authentication is in progress (over an Ethernet interface).

1.5.2.4. Security Management

The TOE provides the VirtualWisdom UI for administrators to manage the security functions, configuration, and other features of the TOE. The Security Management function specifies the administrator defined access for the management of the TOE. The following are TOE management functions provided via the VirtualWisdom Platform Appliance UI:

- Administer the TOE locally and remotely
- Configure the number of unsuccessful authentication attempts for authentication failure management
- Manually update the TOE and verify TOE updates
- Import or delete cryptographic keys
- Configure the login banner
- Configure session timeouts

[11] TSF – TOE Security Functionality
[12] AES – Advanced Encryption Standard
[13] CBC – Cipher Block Chaining mode
[14] GCM – Galois/Counter Mode
[15] RSA – Ron Rivest, Adi Shamir and Leonard Adleman
[16] ECDSA – Elliptic Curve Digital Signature Algorithm
[17] ECDHE – Elliptic Curve Diffie-Hellman
[18] CTR- Counter
[19] DRBG – Deterministic Random Bit Generator

1.5.2.5. Protection of the TSF

The TOE provides reliable timestamps for its own use by synchronizing with an NTP server. Digital signatures are used to verify all firmware updates that are applied to the TOE. The TOE runs a suite of self-tests at power-on and during normal operation to ensure the integrity of the TSF. The TOE hashes passwords and prevents access and reading of plaintext keys and passwords.

1.5.2.6. TOE Access

TOE administrators are automatically logged out of local and remote management interfaces after an administrator-specified amount of idle time. Users can also terminate their own session. The TOE displays an access banner prior to all administrative sessions.

1.5.2.7. Trusted Path/Channels

The TOE provides trusted paths and trusted channels using its cryptographic functions. The TOE secures administrative communications using TLS v1.2 over its management interface.

The TOE also provides trusted TLS v1.2 communications channels between the TOE and the Syslog and NTP servers.

1.5.3. Product Physical/Logical Features and Functionality not included in the TOE

The following features and functionality that are not part of the evaluated configuration of the TOE:

- SSH
- LDAP authentication

1.5.4. Scope of Evaluation

The evaluation is limited in scope to the secure management features described in the *Collaborative Protection Profile for Network Devices* v2.0 + Errata 20180314 and detailed in Logical Scope section of this document.

Chapter 2

Conformance Claims

This section provides the identification for any CC, PP, and EAL package conformance claims. Rationale is provided for any extensions or augmentations to the conformance claims. Rationale for CC and PP conformance claims can be found in Chapter 9. This Security Target also conforms to the applicable NIAP Technical Decisions identified in Table 11 of Chapter 9.

Table 3 – CC and PP Conformance

CC Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 3.1, Release 4, September 2012; CC Part 2 extended; CC Part 3 conformant; PP claim to the collaborative *Protection Profile for Network Devices* v2.0 + Errata 20180314, March 14, 2018 conformant; and no interpretations apply to the claims made in this ST.
PP Identification	Exact Conformance[20] to the collaborative Protection Profile for Network Devices, v2.0 + Errata 20180314.

[20] Exact Conformance is a type of Strict Conformance such that the set of SFRs and the Security Problem Definition/Objectives are exactly as presented within the accepted NDcPP without changes.

Chapter 3

Security Problem Definition

This section describes the security aspects of the environment in which the TOE will be used and the manner in which the TOE is expected to be employed. It provides the statement of the TOE security environment, which identifies and explains all of the following:

- Known and presumed threats countered by either the TOE or by the security environment
- Organizational security policies with which the TOE must comply
- Assumptions about the secure usage of the TOE, including physical, personnel, and connectivity aspects

A network device has a network infrastructure role it is designed to provide. In doing so, the network device communicates with other network devices and other network entities (an entity not defined as a network device) over the network. At the same time, it must provide a minimal set of common security functionality expected by all network devices. The security problem to be addressed by a compliant network device is defined as this set of common security functionality that addresses the threats that are common to network devices, as opposed to those that might be targeting the specific functionality of a specific type of network device. The set of common security functionality addresses communication with the network device, both authorized and unauthorized; the ability to perform valid or secure updates; the ability to audit device activity; the ability to securely store and utilize device and administrator credentials and data; and the ability to self-test critical device components for failures.

3.1. Threats to Security

This section identifies the threats to the IT[21] assets against which the TOE or security environment must provide protection. The threat agents are divided into two categories:

- Attackers who are not TOE users: These threat agents have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings or parameters, and no physical access to the TOE.
- TOE administrative users: These threat agents have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings or parameters, and physical access to the TOE. (It is assumed that TOE administrative users will operate in a trusted manner.)

Both are assumed to have a low level of motivation. The IT assets requiring protection are the TSF and user data saved on the TOE. Removal, diminution, and mitigation of the threats are achieved through the objectives identified in Chapter 4. The section below lists the applicable threats.

The threats for the Network Device are grouped according to functional areas of the device in the sections below.

3.1.1. Communications with the Network Device

A network device communicates with other network devices and other network entities. The endpoints of this communication can be geographically and logically distant and may pass through a variety of other systems. The intermediate systems may be untrusted providing an opportunity for unauthorized communication with the network device or for authorized communication to be compromised. The security functionality of the network device must be able to protect any critical network traffic (administration traffic, authentication traffic, audit traffic, etc.). The communication with the network device falls into two categories: authorized communication and unauthorized communication.

[21] IT – Information Technology

Authorized communication includes network traffic allowable by policy destined to and originating from the network device as it was designed and intended. This includes critical network traffic, such as network device administration and communication with an authentication or audit logging server, which requires a secure channel to protect the communication. The security functionality of the network device includes the capability to ensure that only authorized communications are allowed and the capability to provide a secure channel for critical network traffic. Any other communication is considered unauthorized communication.

The primary threats to network device communications addressed in this cPP focus on an external, unauthorized entity attempting to access, modify, or otherwise disclose the critical network traffic. A poor choice of cryptographic algorithms or the use of non-standardized tunneling protocols along with weak administrator credentials, such as an easily guessable password or use of a default password, will allow a threat agent unauthorized access to the device. Weak or no cryptography provides little to no protection of the traffic allowing a threat agent to read, manipulate and/or control the critical data with little effort. Non-standardized tunneling protocols not only limit the interoperability of the device but lack the assurance and confidence standardization provides through peer review.

3.1.1.1. T.UNAUTHORIZED_ADMISTRATOR_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.

3.1.1.2. 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 or control the traffic with minimal effort.

3.1.1.3. T.UNTRUSTED_COMMUNICATION_CHANNELS

Threat agents may attempt to target network devices that do not use standardized secure tunneling 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 could potentially lead to a compromise of the network device itself.

3.1.1.4. 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 the network device itself could potentially be compromised.

3.1.2. Valid Updates

Updating network device software and firmware is necessary to ensure that the security functionality of the network device is maintained. The source and content of an update to be applied must be validated by cryptographic means; otherwise, an invalid source can write their own firmware or software updates that circumvent the security functionality of the network device. Methods of validating the source and content of a software or firmware update by cryptographic means typically involve cryptographic signature schemes where hashes of the updates are digitally signed.

Unpatched versions of software or firmware leave the network device susceptible to threat agents attempting to circumvent the security functionality using known vulnerabilities. Non-validated updates or updates validated using non-secure or weak cryptography leave the updated software or firmware vulnerable to threat agents attempting to modify the software or firmware to their advantage.

3.1.2.1. 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.

3.1.3. Audited Activity

Auditing of network device activities is a valuable tool for administrators to monitor the status of the device. It provides the means for administrator accountability, security functionality activity reporting, reconstruction of events, and problem analysis. Processing performed in response to device activities may give indications of a failure or compromise of the security functionality. When indications of activity that impact the security functionality are not generated and monitored, it is possible for such activities to occur without administrator awareness. Further, if records are not generated and retained, reconstruction of the network and the ability to understand the extent of any compromise could be negatively affected. Additional concerns are the protection of the audit data that is recorded from alteration or unauthorized deletion. This could occur within the TOE or while the audit data is in transit to an external storage device.

Note this cPP requires that the network device generate the audit data and has the capability to send the audit data to a trusted network entity (e.g., a Syslog server).

3.1.3.1. T.UNDETECTED_ACTIVITY

Threat agents may attempt to access, change, 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.

3.1.4. Administrator and Device Credentials and Data

A network device contains data and credentials which must be securely stored and must appropriately restrict access to authorized entities. Examples include the device firmware, software, configuration authentication credentials for secure channels, and administrator credentials. Device and administrator keys, key material, and authentication credentials need to be protected from unauthorized disclosure and modification. Furthermore, the security functionality of the device needs to require default authentication credentials, such as administrator passwords, be changed.

Lack of secure storage and improper handling of credentials and data, such as unencrypted credentials inside configuration files or access to secure channel session keys, can allow an attacker to not only gain access to the network device, but also compromise the security of the network through seemingly authorized modifications to configuration or though man-in-the-middle attacks. These attacks allow an unauthorized entity to gain access and perform administrative functions using the Security Administrator's credentials and to intercept all traffic as an authorized endpoint. This results in difficulty in detection of security compromise and in reconstruction of the network, potentially allowing continued unauthorized access to administrator and device data.

3.1.4.1. 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 include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.

3.1.4.2. 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.

3.1.5. Device Failure

Security mechanisms of the network device generally build up from roots of trust to more complex sets of mechanisms. Failures could result in a compromise to the security functionality of the device. A network device self-testing its security critical components at both start-up and during run-time ensures the reliability of the device's security functionality.

3.1.5.1. 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

This section describes the assumptions made in identification of the threats and security requirements for network devices. The network device is not expected to provide assurance in any of these areas, and as a result, requirements are not included to mitigate the threats associated.

3.2.1. 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 and/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 will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will 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.

[OE.PHYSICAL]

3.2.2. A.LIMITED_FUNCTIONALITY

The device is assumed to provide networking functionality as its core function and not provide functionality or services that could be deemed as general purpose computing. For example, the device should not provide computing platform for general purpose applications (unrelated to networking functionality).

[OE.NO_GENERAL_PURPOSE]

3.2.3. A.NO_THRU_TRAFFIC_PROTECTION

A standard or 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 and 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 for particular types of network devices (e.g., firewall).

[OE.NO_THRU_TRAFFIC_PROTECTION]

3.2.4. 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 being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords and 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.

[OE.TRUSTED_ADMIN]

3.2.5. 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.

[OE.UPDATES]

3.2.6. 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.

[OE.ADMIN_CREDENTIALS_SECURE]

3.2.7. 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.

[OE.RESIDUAL_INFORMATION]

3.3. Organizational Security Policies

An OCP is a set of rules, practices, and procedures imposed by an organization to address its security needs. For the purposes of this cPP, a single policy is described in the section below.

3.3.1. 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.

[FTA_TAB.1]

Chapter 4

Security Objectives

4.1. Security Objectives for the Operational Environment

The following subsections describe objectives for the Operational Environment.

4.1.1. OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

4.1.2. 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.

4.1.3. 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.

4.1.4. OE.TRUSTED_ADMIN

TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

4.1.5. 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.

4.1.6. 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.

4.1.7. 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.

Chapter 5
Extended Components

This section defines the extended SFRs and extended SARs met by the TOE.

5.1. Extended TOE Security Functional Components

All of the extended requirements in this ST have been drawn from the ND cPP v2.0e. Table 4 identifies all extended SFRs implemented by the TOE. The definitions for these extended SRS are provided in Appendix A of this document.

Table 4 – Extended TOE Security Functional Requirements

Name	Description
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_HTTPS_EXT.1	HTTPS Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSS_EXT.1	TLS Server Protocol
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
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
FPT_SKP_EXT.1	Protection of TSF data (for reading of all symmetric keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_STM_EXT.1	Reliable Time Stamps
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update
FTA_SSL_EXT.1	TSF-initiated session locking

5.2. Extended TOE Security Assurance Components

There are no extended TOE Security Assurance Components.

Chapter 6

Security Assurance Requirements

This cPP identifies the SARs to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

This section lists the set of SARs from CC Part 3 that are required in evaluations against this cPP. Individual Evaluation Activities to be performed are specified in the *Supporting Document, Mandatory Technical Document, Evaluation Activities for Network Device cPP* [SD].

The general model for evaluation of TOEs against STs written to conform to this cPP is as follows: after the ST has been approved for evaluation, the ITSEF[22] will obtain the TOE, supporting environmental IT (if required), and the guidance documentation for the TOE. The ITSEF is expected to perform actions mandated by the Common Evaluation Methodology (CEM) for the ASE and ALC SARs. The ITSEF also performs the Evaluation Activities contained within the SD, which are intended to be an interpretation of the other CEM assurance requirements as they apply to the specific technology instantiated in the TOE. The Evaluation Activities that are captured in the SD also provide clarification as to what the developer needs to provide to demonstrate the TOE is compliant with the cPP.

The TOE security assurance requirements are identified in Table 5.

Table 5 – Security Assurance Requirements

Assurance Requirements	
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life Cycle Support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM[23] coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – sample (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

[22] ITSEF – Information Technology Security Entrepreneurs Forum

[23] CM – Configuration Management

Chapter 7

Security Functional Requirements

The individual security functional requirements are specified in the sections below. SFRs in this section are mandatory SFRs that any conformant TOE must meet. Based on selections made in these SFRs, some of the selection-based SFRs from Appendix B of the NDcPP are also included in the sections below. There are no optional SFRs from those listed in Appendix A of the NDcPP included in this Security Target.

The Evaluation Activities defined in the [SD] describe actions that the evaluator will take in order to determine compliance of a particular TOE with the SFRs. The content of these Evaluation Activities will therefore provide more insight into deliverables required from TOE Developers.

7.1. Conventions

The conventions used in descriptions of the SFRs are as follows:

- Assignment: Indicated with italicized text surrounded by brackets (e.g., [*assignment*]).
- Refinement made in the PP: Indicated with bold text and strikethroughs (e.g., "**refinement**" or "~~refinement~~").
- Selection: Indicated with underlined text surrounded by brackets (e.g., [selection]).
- Assignment within a selection: Indicated with italicized and underlined text surrounded by brackets (e.g., [*assignment within a selection*]);
- Iteration: Indicated by adding a string starting with "/".
- Extended SFRs are identified by having a label 'EXT' at the end of the SFR name.
- Operations such as assignments and selections performed by the PP author are identified as shown above; however, they do not appear within brackets. This is done intentionally to delineate between selections or assignments made by the PP author and those made by the ST author. No refinements have been made by the ST author other than grammatical and formatting corrections, or those made in places where a table reference differs from that of the PP.

7.2. Security Functional Requirements

This section specifies the SFRs for the TOE and organizes the SFRs by CC class. Table 6 identifies all SFRs implemented by the TOE and indicates the ST operations made by the ST author performed on each requirement. Refinements made in the PP are also indicated.

Table 6 – TOE Security Functional Requirements

Name	Description	S	A	R	I
FAU_GEN.1	Audit data generation	✓	✓		
FAU_GEN.2	User identity association				
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/Data Encryption	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)	✓	✓		

Name	Description	S	A	R	I
FCS_HTTPS_EXT.1	HTTPS Protocol	✓			
FCS_RBG_EXT.1	Random Bit Generation	✓	✓		
FCS_TLSC_EXT.1	TLS Client Protocol	✓	✓		
FCS_TLSS_EXT.1	TLS Server Protocol	✓			
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_MOF.1/ ManualUpdate	Management of security functions behavior			✓	
FMT_MTD.1/CoreData	Management of TSF data			✓	
FMT_SMF.1	Specification of management functions	✓	✓		
FMT_SMR.2	Restrictions on Security Roles			✓	
FPT_APW_EXT.1	Protection of Administrator Passwords				
FPT_SKP_EXT.1	Protection of TSF Data (for reading of 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_SSL_EXT.1	TSF-initiated session locking	✓			
FTA_SSL.3	TSF-initiated Termination			✓	
FTA_SSL.4	User-initiated Termination			✓	
FTA_TAB.1	Default TOE access banners			✓	
FTP_ITC.1	Inter-TSF Trust Channel	✓		✓	
FTP_TRP.1/Admin	Trusted Path	✓		✓	

Note: S=Selection; A=Assignment; R=Refinement; I=Iteration

7.2.1. Class FAU: Security Audit

7.2.1.1. FAU_GEN.1 Audit Data Generation

Hierarchical to: No other components
Dependencies: FPT_STM.1 Reliable time stamps

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 7.*

Table 7 – Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	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_TLSC_EXT.1/2	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1/2	Failure to establish a TLS Session	Reason for failure
FCS_RBG_EXT.1	None	None
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded	Origin of the attempt (e.g., IP[24] address)
FIA_PMG_EXT.1	None	None
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address)
FIA_UAU_EXT.2	All use of the 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
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None

[24] IP – Internet Protocol

Requirement	Auditable Events	Additional Audit Record Contents
FMT_MTD.1/CoreData	All management activities of TSF data	None
FMT_SMF.1	None	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
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	No additional information
FTA_SSL_EXT.1	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 attempts
FTP_TRP.1/Admin	Initiation of the trusted path Termination of the trusted path Failure of the trusted path functions	None

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 7.*

7.2.1.2. FAU_GEN.2 User identity association

Hierarchical to: No other components
Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

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.

7.2.1.3. FAU_STG_EXT.1 Protected Audit Event Storage

Hierarchical to:	No other components
Dependencies:	FAU_GEN.1 Audit data generation FTP_ITC.1 Inter-TSF trusted channel

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.

FAU_STG_EXT.1.3
The TSF shall [overwrite previous audit records according to the following rule: [*once there's 10 audit files (each a maximum of 10 MB*[25]*) being used for audit data, the oldest file will be overwritten*]] when the local storage space for audit data is full.

7.2.2. Class FCS: Cryptographic Support

7.2.2.1. FCS_CKM.1 Cryptographic Key Generation

Hierarchical to:	No other components
Dependencies:	FCS_COP.1 Cryptographic operation FCS_CKM.4 Cryptographic key destruction

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[26] PUB[27] 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
- ECC[28] schemes using "NIST curves" [P-256, P-384] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;

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

7.2.2.2. FCS_CKM.2 Cryptographic Key Establishment

Hierarchical to:	No other components
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction

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

[25] Mb – Megabit
[26] FIPS – Federal Information Processing Standard
[27] PUB – Publication
[28] ECC – Elliptic Curve Cryptography

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";

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

7.2.2.3. FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components
Dependencies: FCS_CKM.1 Cryptographic key generation

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* [
  - *logically addresses the storage location of the key and performs a [single-pass] overwrite consisting of [zeros]*];

that meets the following: *No Standard*.

7.2.2.4. FCS_COP.1/Data Encryption Cryptographic Operation (AES Data Encryption/Decryption)

Hierarchical to: No other components
Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

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*[29] *18033-3, [*CBC as specified in ISO 10116, GCM as specified in ISO 19772]].

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

Hierarchical to: No other components
Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

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*]

]

[29] ISO – International Organization for Standardization

that meet the following:
[

- For RSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS[30] #1 v2.1 Signature Schemes RSASSA[31] -PSS[32] and/or RSASSA-PKCS1v1_5; ISO/IEC[33] 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

].

7.2.2.6. FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

Hierarchical to: No other components
Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

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

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

Hierarchical to: No other components
Dependencies: FCS_CKM.1 Cryptographic key generation
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/KeyedHash
The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [HMAC-SHA-256, HMAC-SHA-384] and cryptographic key sizes [*key size (in bits) used in HMAC*] **and message digest sizes [256, 384] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 “MAC[34] Algorithm 2”*.

7.2.2.8. FCS_HTTPS_EXT.1 HTTPS Protocol

Hierarchical to: No other components
Dependencies: FCS_TLS_EXT.1

FCS_HTTPS_EXT.1.1
The TSF shall implement the HTTPS protocol that complies with RFC[35] 2818.

FCS_HTTPS_EXT.1.2
The TSF shall implement HTTPS using TLS.

[30] PKCS – Public Key Cryptography Standard
[31] RSASSA – RSA Signature Scheme with Appendix
[32] PSS – Probabilistic Signature Scheme
[33] IEC – International Electrotechnical Commission
[34] MAC – Message Authentication Code
[35] RFC – Request For Comment

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

7.2.2.9. FCS_TLSC_EXT.1 TLS Client Protocol

Hierarchical to: No other components
Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
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_RBG_EXT.1 Random Bit Generation

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_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

].

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
The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [not establish the connection].

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

7.2.2.10. FCS_TLSS_EXT.1 TLS Server Protocol

Hierarchical to: No other components
Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
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_RBG_EXT.1 Random Bit Generation

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_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

].

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 [generate EC[36] Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, and no other curves]].

7.2.2.11. FCS_RBG_EXT.1 Random Bit Generation

Hierarchical to: No other components
Dependencies: No other components

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[37] shall be seeded by at least one entropy source that accumulates entropy from [[*1*] software-based noise source, [*1*] hardware-based noise source] with 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.

7.2.3. Class FIA: Identification and Authentication

7.2.3.1. FIA_AFL.1 Authentication Failure Management (Refinement)

Hierarchical to: No other components
Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within [*2*-*5*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely*.

FIA_AFL.1.2

[36] EC – Elliptic Curve
[37] RBG – Random Bit Generator

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending remote Administrator from successfully authenticating until an Administrator defined time period has elapsed].

7.2.3.2. FIA_PMG_EXT.1 Password Management

Hierarchical to: No other components
Dependencies: No other components

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [*“`”, ”~”, ”_”, ”+”, ”-“, ”=”, ”{“, ”}”, “|”, “\”, “:”, “””, ”;”, ”’”, “<”, “>”, “?”, “,”, “.”, “/”, “[”, ”]”*]]

b) Minimum password length shall be configurable to [*1*] and [*15 characters*].

7.2.3.3. FIA_UIA_EXT.1 User identification and authentication

Hierarchical to: No other components
Dependencies: FTA_TAB.1 Default TOE Access Banners

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;
- [*Load the VirtualWisdom Platform Appliance UI login page*].

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.

7.2.3.4. FIA_UAU_EXT.2 Password-based Authentication Mechanism

Hierarchical to: No other components
Dependencies: No other components

FIA_UAU_EXT.2.1
The TSF shall provide a local password-based authentication mechanism, and [*no other authentication mechanism*] to perform local administrative user authentication.

7.2.3.5. FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components
Dependencies: FIA_UAU.1 Timing of authentication

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**.

7.2.3.6. FIA_X509_EXT.1/Rev X.509 Certificate Validation

Hierarchical to:	No other components
Dependencies:	No other components

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

- RFC 5280 certificate validation and certificate path validation **supporting a minimum path length of three certificates**.
- The certificate path must terminate with a trusted CA[38] certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- 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*[39] *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.

7.2.3.7. FIA_X509_EXT.2 X.509 Certificate Authentication

Hierarchical to:	No other components
Dependencies:	No other components

FIA_X509_EXT.2.1
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [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].

7.2.3.8. FIA_X509_EXT.3 X.509 Certificate Requests

Hierarchical to:	No other components
Dependencies:	No other components

FIA_X509_EXT.3.1
The TSF shall generate a Certificate Request Message 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].

[38] CA – Certificate Authority

[39] OID – Object Identifier

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

7.2.4. Class FMT: Security Management

7.2.4.1. FMT_MOF.1/ManualUpdate Management of security functions behaviour

Hierarchical to:	No other components
Dependencies:	FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions

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

7.2.4.2. FMT_MTD.1/CoreData Management of TSF data

Hierarchical to:	No other components
Dependencies:	FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles

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

7.2.4.3. FMT_SMF.1 Specification of Management Functions

Hierarchical to:	No other components
Dependencies:	FIA_UIA_EXT.1 User Identification and Authentication FCS_COP.1(2) Cryptographic operation (for cryptographic signature) FPT_TUD_EXT.1 Trusted Update

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*] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [Ability to configure the cryptographic functionality]

7.2.4.4. FMT_SMR.2 Restrictions on security roles

Hierarchical to:	No other components
Dependencies:	FIA_UID.1 Timing of identification

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.

7.2.5. Class FPT: Protection of the TSF

7.2.5.1. FPT_APW_EXT.1 Protection of Administrator Passwords

Hierarchical to:	No other components
Dependencies:	No other components

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

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

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

Hierarchical to:	No other components
Dependencies:	No other components

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

7.2.5.3. FPT_STM_EXT.1 Reliable time stamps

Hierarchical to:	No other components
Dependencies:	No other components

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 [<u>synchronise time with external time sources</u>].

7.2.5.4. FPT_TST_EXT.1 TSF Testing (Extended)

Hierarchical to:	No other components
Dependencies:	No other components

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [<u>during initial start-up (on power on), at the conditions [*execution of DRBG function, generation of asymmetric RSA or ECDSA keypair*]</u>] to demonstrate the correct operation of the TSF:
[

- *Firmware Integrity Test (SHA-256)*
- *Cryptographic Library Integrity Test (HMAC-SHA-256)*
- *Firmware load test with 3072-bit RSA private key*

- *AES ECB[40] KAT[41]*
- *AES GCM KAT*
- *HMAC KAT with SHA-256, SHA-384*
- *SHA KAT with SHA-256, SHA-384*
- *NIST SP[42]800-90A CTR_DRBG KAT*
- *RSA sign/verify KAT*
- *ECDSA sign/verify KAT*
- SP 800-56A Primitive "Z" Computation KAT
- *Pairwise Consistency Test (PCT) for RSA keypairs*
- *ECDSA PCT*
- *DRBG Health Checks: instantiate, uninstantiate, generate, and reseed*

].

7.2.5.5. FPT_TUD_EXT.1 Trusted Update

Hierarchical to:	No other components
Dependencies:	FCS_COP.1/SigGen Cryptographic operation (for Cryptographic Signature and Verification), or FCS_COP.1/Hash Cryptographic operation (for cryptographic hashing)

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 [the most recently installed version of the TOE firmware/software].

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 mechanism] prior to installing those updates.

7.2.6. Class FTA: TOE Access

7.2.6.1. FTA_SSL_EXT.1 TSF-initiated session locking

Hierarchical to:	No other components
Dependencies:	No other components

FTA_SSL_EXT.1.1
The TSF shall, for local interactive sessions, [terminate the session] after a Security Administrator-specified time period of inactivity.

7.2.6.2. FTA_SSL.3 TSF-initiated Termination (Refinement)

Hierarchical to:	No other components
Dependencies:	No other components

FTA_SSL.3.1

[40] ECB – Electronic Codebook

[41] KAT – Known Answer Test

[42] SP – Special Publication

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

7.2.6.3. FTA_SSL.4 User-initiated Termination (Refinement)

Hierarchical to: No other components
Dependencies: No other components

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

7.2.6.4. FTA_TAB.1 Default TOE access banners (Refinement)

Hierarchical to: No other components
Dependencies: No other components

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.

7.2.7. Class FTP: Trusted Path/Channels

7.2.7.1. FTP_ITC.1 Inter-TSF trusted channel (Refinement)

Hierarchical to: No other components
Dependencies: No other components

FTP_ITC.1.1
The TSF shall be **capable of using [<u>TLS</u>] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [<u>*NTP server*, no other capabilities</u>]** 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 **<u>the TSF or the authorized IT entities</u>** to initiate communication via the trusted channel.

FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for [*sending audit data*, *synchronizing time to an external time source*].

7.2.7.2. FTP_TRP.1/Admin Trusted path (Refinement)

Hierarchical to: No other components
Dependencies: No other components

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

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*.

Chapter 8

TOE Summary Specification

This section presents information to detail how the TOE meets the functional requirements described in previous sections of this ST.

8.1. TOE Security Functionality

Each of the security requirements and the associated descriptions correspond to the security functions. Hence, each function is described by how it specifically satisfies each of its related requirements. This serves to both describe the security functions and rationalize that the security functions satisfy the necessary requirements.

Table 8 – Mapping of TOE Security Functionality to Security Functional Requirements

TOE Security Function	SFR ID[43]	Description
Security Audit	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_STG_EXT.1	Protected audit trail storage
Cryptographic Support	FCS_CKM.1	Cryptographic Key Generation
	FCS_CKM.2	Cryptographic Key Establishment
	FCS_CKM.4	Cryptographic Key Destruction
	FCS_COP.1/Data Encryption	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
	FCS_TLSS_EXT.1	TLS Server Protocol
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
Security Management	FMT_MOF.1/ManualUpdate	Management of security functions behavior
	FMT_MTD.1/CoreData	Management of TSF data
	FMT_SMF.1	Specification of management functions
	FMT_SMR.2	Restrictions on Security Roles
Protection of the TSF	FPT_APW_EXT.1	Protection of Administrator Passwords

[43] ID – Identifier

TOE Security Function	SFR ID[43]	Description
	FPT_SKP_EXT.1	Protection of TSF Data (for reading of 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
TOE Access	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 banners
Trusted path/channels	FTP_ITC.1	Inter-TSF Trust Channel
	FTP_TRP.1/Admin	Trusted Path

8.1.1. Security Audit

The Security Audit function provides the TOE with the functionality of generating audit records. As administrators manage and configure the TOE, their activities are tracked and recorded as audit records and are stored in the file system. The resulting audit records can be examined to determine which security relevant activities took place and who (i.e., which user) is responsible for those activities. The TOE generates audit records for all the required events specified in Table 7, the start-up and shut-down of the audit functions, and the following administrative actions:

- Administrative login and logout
- Changes to TSF data related to configuration changes
- Generating/import of, changing, or deleting of cryptographic keys
- Resetting passwords

The audit records include the following information:

- Date and time of the event
- Type of event
- Subject identity
- Outcome (success or failure) of the event
- Reason for failure to establish HTTPS or TLS sessions
- Origin of the attempt (IP address) of all use of the identification and authentication mechanism
- Reason for failure to validate a certificate
- Identification of the initiator and target of failed trusted channels establishment attempts
- Identity of the user (for user-initiated events)
- Unique key name or key reference associated with keys generated, imported, changed, or deleted. The unique key name or reference is the subject field contents of the certificate associated with the key.
- User accounts associated with administrative login and logout or the resetting of passwords

Audit start-up and shut-down is tied to system start-up and shutdown. The audit functions starts when the system starts, which is indicated by an event in the VirtualWisdom UI audit log. The audit functions shuts down when the system shuts down, which is also indicated by an event in the VirtualWisdom UI audit log.

The TOE stores up to 100 MB of audit data in its local storage space. It writes audit data to 10 log files. When a log file reaches 10 MB, it creates a new log file. After 10 log files have been written, the TOE overwrites previous audit records, beginning with the oldest log file. Authorized administrators may download the local audit files for viewing.

The TOE transmits audit data to an external Syslog server over a TLS v1.2 trusted channel (refer to the TLS Client Protocol section below). The transmission is done in real-time.

TOE Security Functional Requirements Satisfied: FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1.

8.1.2. Cryptographic Support

The TOE provides cryptographic algorithms for key generation, establishment, and destruction; AES encryption and decryption; signature generation and verification; hash and keyed hash generation; and random bit generation. These algorithms support the TOE's implementation of HTTPS and TLS v1.2 for trusted path and channel communications.

8.1.2.1. Cryptographic Algorithms

The following CAVP[44] validated cryptographic algorithms are provided by the TOE using the SLES[45] 12 OpenSSL cryptographic module:

- AES CBC (Cert.# 5528)
- AES CBC AESNI Enabled (Cert. #5508)
- AES GCM (Cert. #5528)
- AES GCM AESNI Enabled (Cert. #5508)
- RSA (Cert. #2965)
- ECDSA (Cert. #1486)
- SHS (Cert. #4436)
- HMAC (Cert. #3681)
- CTR_CRBG (Cert. #2189)
- CTR_DRBG AESNI Enabled (Cert. #2178)KAS ECC CVL[46] (Cert. #1971)

The TOE generates RSA keys (2048 or 3072 bits) and ECDSA keys (using curve P-256 and P-384) according to FIPS PUB 186-4. Key establishment is performed using elliptic curve-based schemes that meet NIST SP 800-56A Revision 2. Key establishment is used when the TOE is the sender (NTP and Syslog servers) and recipient (VirtualWisdom UI).

The TOE performs AES encryption and decryption for HTTPS and TLS v1.2 trusted path and channel communications. The AES algorithm operates in CBC and GCM modes with key sizes of 128 and 256 bits. In TLS sessions, the TOE acts as a TLS Server for the VirtualWisdom Platform Appliance UI and a TLS Client for NTP and Syslog. Please refer to the TLS Client Protocol and TLS Server Protocol sections below for more information on the implementation of the TLS protocols, including whether or not client authentication is supported or performed by the TOE.

For signature generation and verification, the TOE uses RSA and ECDSA algorithms with keys that are greater than or equal to 2048 or 256 bits, respectively. The RSA algorithm meets FIPS PUB

[44] CAVP – Cryptographic Algorithm Validation Program

[45] SLES – SUSE Linux Enterprise Server

[46] CVL – Component Validation List

186-4, Section 5.5, using PKCS #1 v1.5; the ECDSA algorithm meets FIPS PUB 186-4, Section 6 and Appendix D, with NIST curves P-256 and P-384. RSA 3072 with SHA-256 digital signatures are used for firmware upgrades and RSA 2048 or 3072 with HMAC SHA-256 or HMAC-SHA-384 are used for TLS certificates. ECDSA with HMAC-SHA-256 or HMAC-SHA-384 is also used for TLS certificates.

SHA-2 hashing services are performed by the TOE with message digest sizes of 256 and 384. The hash functions are used with other TSF cryptographic functions, including digital signature verification and MACs. The HMAC-SHA-256 cryptographic algorithm uses the SHA-256 hash function with a cryptographic key size of 256 bits and 256 bit message digest size in accordance with the ISO/IEC 9797-2:2011, section 7 “MAC Algorithm 2”. The HMAC-SHA-384 cryptographic algorithm uses the SHA-384 hash function with a cryptographic key size of 384 bits and 384 bit message digest size in accordance with the ISO/IEC 9797-2:2011, section 7 “MAC Algorithm 2”. The values used by the HMAC functions are shown in Table 9.

Table 9 – HMAC Function Values

HMAC	Key Length	Block Size	Output Length
HMAC-SHA-256	256-bit	512-bit	256-bit
HMAC-SHA-384	384-bit	1024-bit	384-bit

All deterministic RBG services performed by the TOE use a 256 bit CTR_DRBG that is seeded with a minimum of 256 bits of entropy by an entropy source (/dev/random) that accumulates entropy from one hardware and one software-based noise source. Hardware entropy is gathered from the RDRAND[47] instruction on the Intel Xeon E5-2680 CPU[48] chip located within the VirtualWisdom Platform Appliance. The RDRAND instruction relies on the underlying built-in DRNG[49] of the Intel Xeon E5-2680 CPU chip, which uses thermal noise within the silicon to output a random stream of bits. Software entropy is collected from Linux kernel events of the VirtualWisdom Platform Appliance. Since there is no way to access the raw noise data produced by the Intel DRNG, it is assumed that the DRNG produces random numbers that are fully entropic (i.e., 8 bits of entropy per byte).

The TOE destroys plaintext keys in volatile storage (RAM) by nulling out the reference to the key (updating the key object with zeros) and then making a request for garbage collection. The Java garbage collection mechanism that is used clears the memory occupied by the key and then releases this memory back to the TOE for reuse. Keys stored in non-volatile memory (file system) are destroyed by logically addressing the storage location of the key and performing a single-pass overwrite consisting of zeros. This is performed with the Linux *shred* command. The RSA 3072-bit update key stored on the device for verifying firmware upgrades cannot be zeroized. Refer to Table 10 for more information on the relevant keys and key destruction methods.

Table 10 – Key Destruction

Key	Description	Destruction Method
TLS Server RSA Private Keys	Used for digital signatures within the TLS protocol.	Stored in non-volatile memory. Destroyed at factory reset or deletion of the certificate by overwriting with zeros.

[47] RDRAND – RDRAND is an instruction for returning random numbers from an Intel on-chip hardware random number generator.

[48] CPU – Central Processing Unit

[49] DRNG – Digital Random Number Generator

Key	Description	Destruction Method
ECC DH Private and Public Components	Used for key agreement within the TLS protocol.	Stored in volatile memory and destroyed at the end of the TLS session or by power-cycling by updating the key object with zeros and then making a request for garbage collection.
TLS Pre-Master Secret	Pre-Master secret for TLS. Established using ECC DH key agreement.	Stored in volatile memory and destroyed at the end of the TLS session or by power-cycling by updating the key object with zeros and then making a request for garbage collection.
TLS Master Secret	Master secret for TLS. Derived from TLS Pre-Master secret	Stored in volatile memory and destroyed at the end of the TLS session or by power-cycling by updating the key object with zeros and then making a request for garbage collection.
TLS Authentication Key	For HMAC-SHA-256 and HMAC-SHA-384	Stored in volatile memory and destroyed at the end of the TLS session or by power-cycling by updating the key object with zeros and then making a request for garbage collection.
TLS Session Key	AES (CBC or GCM) key	Stored in volatile memory and destroyed at the end of the TLS session or by power-cycling by updating the key object with zeros and then making a request for garbage collection.
Administrator Password	This is a variable 15+ character password that is used to authenticate Administrators	Stored via SHA-256 hash in non-volatile memory and zeroized by overwriting with zeros (if deleted) or new password (if modified).
DRBG Entropy	Generated internally via /dev/random	Stored in volatile memory and destroyed at the end of seeding function or power cycle of the device by updating the key object with zeros and then making a request for garbage collection.
DRBG Seed	Generated using the DRBG entropy	Stored in volatile memory and destroyed at the end of seeding function or power cycle of the device by updating the key object with zeros and then making a request for garbage collection.

8.1.2.2. TLS Client Protocol

The TOE enforces client-side TLS v1.2 for connections to an external NTP and Syslog server. The client-side TLS connections support the following ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

The TOE shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6. The TOE does not establish the connection to the NTP or Syslog server if the server certificate is invalid. For all three servers, the TOE establishes DNS names in the Common Name for the Subject Name field and the Subject Alternative Name field as acceptable reference identifiers. For certificate verification, the TOE compares the acceptable reference identifiers to the identifiers in the presented TLS certificate. The presented identifiers have to match the reference identifier in order to establish the connection.

Certificate pinning is not supported or used by the TOE. The TOE follows best practices regarding matching if a certificate with wildcards is presented.

The TSF generates ephemeral elliptic curve key agreement parameters for the server Key Exchange message using EC[50] Diffie-Hellman parameters over NIST curves secp256r1 and secp384r1. The Supported Elliptic Curves extension is performed by default and not configured by an Administrator.

8.1.2.3. TLS Server Protocol

The TOE supports server-side TLS v1.2 for secure connections from the remote management workstation to the VirtualWisdom Platform Appliance UI (HTTPS). The server-side TLS v1.2 connections support the following ciphersuites:

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

The TOE only accepts TLS v1.2 requests and denies any other SSL or TLS connection requests. Authorized administrators can import TLS server certificates that are signed by an external CA. Imported certificates are not accepted unless the RSA key size is 2048 bits. If a non-supported custom certificate key size is uploaded, an error message will pop up that indicates that the key size is invalid.

The TSF generates ephemeral elliptic curve key agreement parameters for the server Key Exchange message using EC Diffie-Hellman parameters over NIST curves secp256r1 and secp384r1.

8.1.2.4. HTTPS Protocol

The TOE implements HTTPS on trusted paths in compliance with RFC 2818. Acting as a server during remote administration TLS connections, the TOE requires the peer to initiate the connection and does not enforce mutual certificate-based authentication; that is, it does not send a client certificate request, and if a certificate is presented, it will not require client authentication.

TOE Security Functional Requirements Satisfied: FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1KeyedHash, FCS_HTTPS_EXT.1, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1, FCS_RBG_EXT.1.

8.1.3. Identification and Authentication

The TOE provides a password-based logon mechanism, giving Administrators the means to compose a strong password. Administrators can configure passwords to be at least a minimum

[50] EC – Elliptic Curve

password length of 15 characters. Valid passwords can be composed of any combination of upper and lower case letters, numbers, and special characters. The special characters can be any of the following printable characters: ! @ # $ % ^ & * ( ) ` ~ _ + - = { } | \ : ” ; ‘ < > ? , . / [ ].

The VirtualWisdom Platform Appliance UI can be accessed locally via a direct Ethernet connection to the VirtualWisdom Platform Appliance appliance management port or remotely via secure TLS v1.2 connections. The direct Ethernet connection is used for local interactive management. There is no local console access to the VirtualWisdom appliance. In both cases (local and remote management), the VirtualWisdom Platform Appliance UI Logon page is accessed by typing the VirtualWisdom Platform Appliance IP address into a supported web browser. To avoid attacks where an attacker might observe a password being typed by an administrator, passwords are obscured with bullets during local logon to the VirtualWisdom UI.

Administrators logon to the TOE both locally and remotely with a username and password. If these credentials match those stored locally, the administrator is successfully logged on to the TOE. The only actions a local or remote administrator can perform prior to logging on are viewing the TOE access banner and loading the VirtualWisdom UI Login page.

The TOE manages authentication failures by detecting when an administrator-configurable number from two-five of successive unsuccessful remote authentication attempts are made and preventing the offending remote Administrator from further logon attempts until an administrator defined time period has elapsed. The TOE uses N-1 logic, with N being the administrator-configurable number of authentication attempts. For example, if N is configured to two, the user will be locked out after the first bad attempt. The TOE recognizes when the VirtualWisdom UI is being used for local interactive management and does not subject local Administrator logon to blocking (i.e., it does not apply the timeout).

The TOE uses X.509v3 certificates as defined by RFC 5280 when it acts as a TLS client for NTP and Syslog server TLS v1.2 authentication. The TOE validates certificates according to the following rules:

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of three certificates.
- The certificate path must terminate with a trusted CA[51] certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3. Revocation checks are performed when an NTP or Syslog certificate is used in an authentication step.

The TOE only treats a certificate as a CA certificate if the basicConstraints extension is presented and the CA flag is set to TRUE. When an otherwise valid NTP or Syslog server X.509 certificate lacks the Server Authentication purpose in the extendedKeyUsage field, the TOE will reject the certificate and the connection is not established. The TOE does not perform any other checks of the extendedKeyUsage field (i.e., Code Signing, Client Authentication, or OCSP signing purpose).

The TOE uses the CRL Distribution Point (CDP) field of a server certificate for locating CRLs. It checks for an updatedCRL on each connection. If a CRL cannot be updated dynamically to complete the revocation check, the TOE accepts the certificate. The TOE checks that the cRLSign key usage bit is set in certificates that are used to verify signatures on CRLs. The TOE also checks expiration dates of certificates by checking the notAfter and notBefore dates.

[51] CA – Certificate Authority

The TOE generates CRMs that can be sent to a CA to be transformed into an X.509 certificate. To do so, it first generates an RSA 2048 bit key pair and uses the corresponding private key to sign the CRM. The CRMs are specified by RFC 2986 and provide the following information in the request: public key and Common Name (CN), Organization (O), Organizational Unit (OU), Country (C). The TOE validates the chain of certificates from the Root CA upon receiving the CA Certificate Response.

TOE Security Functional Requirements Satisfied: FIA_AFL_ FIA_PMG_EXT.1, FIA_UIA_EXT.1, FIA_UAU_EXT.2, FIA_UAU.7, FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3.

8.1.4. Security Management

Security management specifies how the TOE manages several aspects of the TSF, including TSF data and security functions. TSF data includes configuration data of the TSF and audit data. The TOE provides authorized administrators at the VirtualWisdom UI with the ability to easily manage the security functions and TSF data of the TOE. There are no administrative functions accessible through the VirtualWisdom UI prior to administrator login.

The TOE maintains the Administrator role. The Administrator role has the "Security Administrator" privileges discussed throughout the ST and is responsible for all Security Administrator restricted management functions. The Administrator role can manage the TOE locally and remotely. Management functions the Security Administrator can perform remotely include configuring the TOE access banner, authentication failure parameters, and session inactivity timeouts; initiating and verifying manual TOE updates; downloading the local audit log file; and generating, importing, or deleting cryptographic keys. Locally, the Security Administrator can perform network setup, change the language, set the timezone, and configure NTP.

TOE Security Functional Requirements Satisfied: FMT_MOF.1/ManualUpdate, FMT_MTD.1/CoreData, FMT_SMF.1, FMT_SMR.2.

8.1.5. Protection of the TSF

This section defines requirements for the TOE to protect critical security data such as keys and passwords, to provide self-tests that monitor continued correct operation of the TOE (including detection of failures of firmware or software integrity), and to provide trusted methods for updates to the TOE firmware/software. In addition, the TOE is required to provide reliable timestamps in order to support accurate audit recording under the FAU_GEN family.

Administrator passwords used for login to the VW UI (either locally or remotely) are hashed with SHA-256, and then stored within the non-volatile hard drive of the VirtualWisdom Platform Appliance file system. There is no interface provided for specifically reading the password or password file such that the passwords are displayed in plain text. In addition, all persistently stored private keys listed in the Cryptographic Support section are stored in plaintext keystores within the non-volatile hard drive of the TOE files system. File system keystores are inaccessible to users and cannot be viewed via any management interface. The plaintext keys within the volatile memory (see the Cryptographic Support section) cannot be viewed via the management interfaces and are destroyed after use.

The TOE provides reliable time stamps for the Security Audit functionality, for tracking the inactivity of administrative sessions, and for cryptographic functions. An external NTP server can be used to synchronize the VirtualWisdom clock and provide reliable time stamps for Syslog messages.

The current TOE software version is displayed on the VirtualWisdom UI **Settings → Software Upgrade** page. Only Administrators can manually initiate updates to the TOE by using the **Settings → Software Upgrade** menu to upload a software update. The TOE authenticates updates using an RSA 3072-bit digital signature mechanism prior to allowing an administrator to install them (i.e., apply the update). The TOE does not support delayed activation of software updates.

Digitally signed update bundles are acquired by contacting Virtual Instruments Technical Support. When an Administrator performs the software upgrade, the TOE performs a Firmware Load test. The Firmware Load test validates the update bundle by verifying the digital signature with an RSA 3072-bit update key hardcoded in the appliance. If the digital signature is not verified, the software upgrade cannot proceed. If the digital signature is verified, an administrator can apply the update to Virtual Wisdom, at which point the TOE software extracts the files from the update bundle and performs the software upgrade. After the software is installed, the TOE is automatically restarted and requires the Administrator to log in again for the upgrade process to be complete. If the newly installed software fails for any reason, the Security Administrator must try the software download again or contact Virtual Instruments Technical Support.

The TOE performs a suite of FIPS power-up and conditional self-tests to verify its correct operation. If any of these self-tests fail, the TOE enters into a critical error state and the appliance must be rebooted by an Administrator to run the tests again. These tests are sufficient to validate the correct operation of the TSF because they verify that the TOE firmware/software has not been tampered with and that the cryptographic operations are all performing as expected. The following is a description of each of the start-up tests:

- Firmware Integrity Test: The module verifies the integrity of the entire TOE firmware/software image using SHA-256 checksums during the first phase of the boot process. If the SHA-256 checksums are verified (the newly-computed SHA-256 checksums match the stored checksums), the test is passed and the boot process proceeds. If the test fails, the module enters a critical error state and halts the boot process. This test meets the FIPS PUB 140-2, Security Requirements for Cryptographic Modules (section 4.9.1), which requires that the software/firmware integrity test uses an EDC[52] or an approved authentication technique to test the integrity of software/firmware components when the TOE is powered up.

- Kernel and Cryptographic Library Integrity Test (HMAC-SHA-2): The TOE performs an HMAC-SHA-256 verification of the Linux kernel and the OpenSSL cryptographic libraries.

- AES ECB encrypt/decrypt KAT: The AES ECB KAT takes a known 256-bit key and plaintext value, encrypts the plaintext value, and compares the resulting ciphertext value to the expected ciphertext value to test that the encryption operation is working correctly. If the values differ, the test is failed. The AES ECB KAT then reverses this process by taking the ciphertext value and key, performing decryption, and comparing the result to the known plaintext value to test that the decrypt operation is working correctly. If the values differ, the test is failed. If they are the same, the test is passed. This test meets the FIPS PUB 140-2, Security Requirements for Cryptographic Modules (section 4.9.1), which requires that a cryptographic algorithm test using a KAT shall be conducted for all

[52] EDC – Error Detection Code

cryptographic functions of each Approved cryptographic algorithm implemented by the TOE.

- HMAC KAT with SHA-256, SHA-384: The HMAC implementation creates a MAC using known input data and a known key. This MAC value is then compared to the expected MAC value to test that the HMAC and hash operations are working correctly. If the values differ, the test fails. If they are the same, the test passes. This test meets the FIPS PUB 140-2, Security Requirements for Cryptographic Modules (section 4.9.1), which requires that a cryptographic algorithm test using a KAT shall be conducted for all cryptographic functions of each Approved cryptographic algorithm implemented by the TOE.

- SHA KAT with SHA-256, SHA-384: The SHA implementation is further tested in a SHA KAT. Again, a known input data is used and a hash is created out of the input data. This hash is compared to the expected hash to check that the hash operation is working correctly. If the values differ, the test fails. If they are the same, the test passes. This test meets the FIPS PUB 140-2, Security Requirements for Cryptographic Modules (section 4.9.1), which requires that a cryptographic algorithm test using a KAT shall be conducted for all cryptographic functions of each Approved cryptographic algorithm implemented by the TOE.

- NIST SP 800-90A CTR DRBG: Known values are used to seed and initialize the DRBG. A block of random data is then generated by the DRBG and compared to a value pre-generated using the same known values to test that the DRBG is working correctly. If the random data blocks are the same, the test is passed. Otherwise, it is failed. This test meets the FIPS PUB 140-2, Security Requirements for Cryptographic Modules (section 4.9.1), which requires that a cryptographic algorithm test using a KAT shall be conducted for all cryptographic functions of each Approved cryptographic algorithm implemented by the TOE.

- RSA signature generation/verification KAT: A known private 2048-bit key (with SHA-256) is used to sign a known block of data, and the resultant value is compared with the expected ciphertext to check that the encrypt operation is working correctly. If they differ, the test fails. If they are the same, then the public key is used to decrypt the ciphertext and the output is compared to the original data to test that the decrypt operation is working correctly. If they are the same, the test passes. Otherwise, it is failed. This test meets the FIPS PUB 140-2, Security Requirements for Cryptographic Modules (section 4.9.1), which requires that a cryptographic algorithm test using a KAT shall be conducted for all cryptographic functions of each Approved cryptographic algorithm implemented by the TOE.

- ECDSA signature generation/verification KAT (P-224): The private key is used to sign a block of data, and the resultant value is compared with the original data. If they are the same, the test fails. If they differ, then the public key is used to verify the ciphertext and the output is compared to the original data. If they are the same, the test passes. Otherwise, it fails.

- SP 800-56A Primitive “Z” Computation KAT: The value of the x-coordinate is computed. If it matches its expected value, the test passes. Otherwise, the test fails.

In addition, the following conditional tests are performed during normal operation of the TOE:

- PCT for RSA keypairs: This test is activated whenever an asymmetric RSA keypair is generated by the module to verify that the RSA key agreement functions are working correctly. The RSA private key is used to sign a block of data. The resulting signature is compared to the original data before it was signed. If the two values are equal, then the test fails. If the two values differ, the RSA public key is used to verify the signature and the resulting value is compared to the original data. If they are the same, the test is passed. Otherwise, it is failed.

- ECDSA PCT for key generation: The ECDSA private key is used to sign a block of data. The resulting signature is compared to the original data before it was signed. If the two values are equal, then the test fails. If the two values differ, the ECDSA public key is used to verify the signature and the resulting value is compared to the original data. If they are the same, the test passes. Otherwise, it fails.

- Firmware Load Test: Prior to upgrading the TOE firmware/software image, the module verifies that the upgrade package is properly signed by verifying the signature against an externally calculated signature. The verification uses a 3072-bit RSA key with SHA256 digest.

- Continuous Random Number Generator Test for DRBG: This test is activated on the DRBG implementation whenever a fresh random value is requested. The new random number returned from the DRBG will be compared with the previous random number from the same DRBG to determine if stuck-at-constant type of failure is occurring. This test meets the FIPS PUB 140-2, Security Requirements for Cryptographic Modules (section 4.9.2).

TOE Security Functional Requirements Satisfied: FPT_APW_EXT.1, FPT_SKP_EXT.1, FPT_STM_EXT.1, FPT_TST_EXT.1, FPT_TUD_EXT.1

8.1.6. TOE Access

Using the VirtualWisdom UI, the TOE can be accessed remotely via TLS v1.2 connections or locally through a direct Ethernet connection from the management workstation to the management port of the VirtualWisdom Platform Appliance. Before an administrative session can be established either locally or remotely, the VirtualWisdom UI displays a login banner (configured by a Security Administrator) warning against unauthorized use of the TOE.

Session termination is implemented to mitigate the risk of an account being used illegitimately. The VirtualWisdom UI allows Administrators to terminate their own interactive sessions by logging out. Otherwise, the VirtualWisdom UI terminates both local and remote interactive sessions after a specified time period of inactivity configured by a Security Administrator.

TOE Security Functional Requirements Satisfied: FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_TAB.1.

8.1.7. Trusted Path/Channels

All secure channels provided by the TOE conform to the TLS requirements in the Class FCS: Cryptographic Support section. The TOE communicates with authorized external IT entities via the following secure channel:

- TLS v1.2 for connections to the Syslog and NTP servers

The TOE provides trusted paths via HTTPS over TLS v1.2 for connections to the VirtualWisdom UI. The trusted paths provided by the TOE conform to the TLS requirements in the Class FCS: Cryptographic Support section.

TOE Security Functional Requirements Satisfied: FTP_ITC.1, FTP_TRP.1/Admin.

Chapter 9

Rationale

9.1. Conformance Claims Rationale

This Security Target extends Part 2 and conforms to Part 3 of the Common Criteria Standard for Information Technology Security Evaluations, Version 3.1 Release 4. This ST conforms to the ND cPP, including the mandatory SFRs as well as the following selection-based SFRs. No optional SFRs are implemented by the TOE.

- FCS_HTTPS_EXT.1
- FCS_TLSC_EXT.1
- FCS_TLSS_EXT.1
- FIA_X509_EXT.1/Rev
- FIA_X509_EXT.2
- FIA_X509_EXT.3

This Security Target also conforms to the following applicable NIAP Technical Decisions:

Table 11 – NIAP Technical Decisions

Item	Description
TD0228	NIT[53] Technical Decision for CA certificates – basicConstraints validation
TD0256	NIT Technical Decision for Handling of TLS connections with and without mutual authentication
TD0257	NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/FCS_TLSC_EXT.x.2 Tests 1-4
TD0289	NIT Technical Decision for FCS_TLSC_EXT.X.1 Test 5e
TD0290	NIT Technical Decision for Physical Interruptions of Trusted Path/Channel
TD0291	NIT technical decision for DH14 and FCS_CKM.1
TD0321	Protection of NTP communications
TD0324	NIT Technical Decision for Correction of section numbers in SD Table 1
TD0333	NIT Technical Decision for Applicability of FIA_X509_EXT.3
TD0338	NIT Technical Decision for Access Banner Verification
TD0340	NIT Technical Decision for Handling of the basicConstraints extension in CA and leaf certificates
TD0341	NIT Technical Decision for TLS wildcard checking
TD0342	NIT Technical Decision for TLS and DTLS Server Tests

9.1.1. Variance Between the PP and this ST

There is no variance between the ND cPP and this ST.

9.1.2. Security Assurance Requirements Rationale

This ST claims exact conformance to the ND cPP, including the assurance requirements listed in Section 6 of the ND cPP.

9.1.3. Dependency Rationale

The SFRs in this Security Target represent the SFRs identified in the ND cPP v2.0 + Errata 20180314. As such, the ND cPP v2.0 + Errata 20180314 SFR dependency rationale is deemed acceptable since the PP itself has been validated.

[53] NIT – Network Interpretations Team

Chapter 10

Acronyms and Terms

Table 12 and Table 13 describe the acronyms and terms used throughout the document.

10.1. Acronyms

Table 12 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CA	Certificate Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining Mode
CC	Common Criteria
CDP	Certificate Validation List Distribution Point
CEM	Common Evaluation Methodology
CM	Configuration Management
CN	Common Name
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRL	Certificate Revocation List
CRM	Certificate Request Message
CVL	Component Validation List
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DRNG	Digital Random Number Generator
DSS	Digital Signature Standard
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography
ECDHE	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EDC	Error Detection Code
FIPS	Federal Information Processing Standard
GCM	Galois/Counter Mode
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
ID	Identifier
IEC	International Electrotechnical Commission
I/O	Input/Output
IP	Internet Protocol
IPM	Infrastructure Performance Management

Acronym	Definition
IRQ	Interrupt Request Channel
ISO	International Organization for Standardization
IT	Information Technology
ITSEF	Information Technology Security Entrepreneurs Forum
KAT	Known Answer Test
MAC	Message Authentication Code
NAS	Network-Attached Storage
ND cPP	collaborative Protection Profile for Network Devices v2.0
NDRNG	Non-Deterministic Random Number Generator
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NIT	Network Interpretations Team
NTP	Network Time Protocol
O	Organization
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OSP	Organizational Security Policy
OU	Organizational Unit
PCT	Pairwise Consistency Test
PKCS	Public Key Cryptography Standard
PP	Protection Profile
PSS	Probabilistic Signature Scheme
PUB	Publication
RBG	Random Bit Generator
RFC	Request For Comment
RSA	Ron Rivest, Adi Shamir and Leonard Adleman
RSASSA	RSA Signature Scheme with Appendix
SAN	Storage Area Network
SAR	Security Assurance Requirement
SD	Supporting Document
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SP	Special Publication
SSL	Secure Sockets Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation

Acronym	Definition
TSF	TOE Security Functionality
UI	User Interface
VM	Virtual Machine

10.2. Terms

Table 13 – Terms

Name	Definition
Administrator	See Security Administrator.
Assurance	Grounds for confidence that a TOE meets the SFRs.
Security Administrator	The terms “Administrator” and “Security Administrator” are used interchangeably in this document.
TSF Data	Data for the operation of the TSF upon which the enforcement of the requirements relies.

Appendix A:

Extended Component Definitions

This appendix contains the definitions for the extended requirements that are used in this Security Target. (Note: formatting conventions for selections and assignments in this Appendix are those in [CC2].)

A.1 Class FAU: Security Audit

Families in this class address the requirements for functions to implement security audit as defined in CC Part 2.

A.1.1 Protected audit event storage (FAU_STG_EXT)

A.1.1.1 Family Behavior

This component defines the requirements for the TSF to be able to securely transmit audit data between the TOE and an external IT entity.

A.1.1.2 Component Leveling

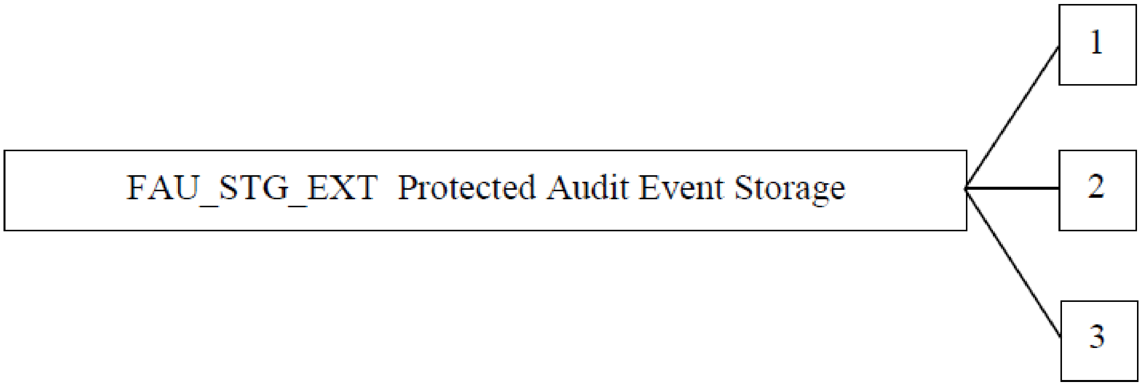

Figure 3 – Protected Audit Event Storage family decomposition

FAU_STG_EXT.1 Protected audit event storage requires the TSF to use a trusted channel implementing a secure protocol.

A.1.1.3 Management: FAU_STG_EXT.1

The following actions could be considered for the management functions in FMT:

a) The TSF shall have the ability to configure the cryptographic functionality.

A.1.1.4 Audit: FAU_STG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

A.1.1.5 FAU_STG_EXT.1 Protected Audit Event Storage

Hierarchical to:	No other components
Dependencies:	FAU_GEN.1 Audit data generation FTP_ITC.1 Inter-TSF trusted channel

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.

FAU_STG_EXT.1.2

The TSF shall be able to store generated audit data on the TOE itself.

FAU_STG_EXT.1.3
The TSF shall [selection: *drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]*] when the local storage space for audit data is full.

A.2 Class FCS: Cryptographic Support

Families in this class address the requirements for functions to implement cryptographic functionality as defined in CC Part 2.

A.2.1 Random Bit Generation (FCS_RBG_EXT)

A.2.1.1 Family Behavior

Components in this family address the requirements for random number / bit generation. This is a new family defined for the FCS Class.

A.2.1.2 Component Leveling

Figure 4 – Random Bit Generation family decomposition

FCS_RBG_EXT.1 Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

A.2.1.3 Management: FCS_RBG_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

A.2.1.4 Audit: FCS_RBG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: failure of the randomization process.

A.2.1.5 FCS_RBG_EXT.1 Random Bit Generation

Hierarchical to:	No other components.
Dependencies:	No dependencies.

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

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: *[assignment: number of software-based sources] software-based noise source, [assignment: number of hardware-based sources] hardware-based noise source*] with minimum of [selection: *128 bits, 192 bits, 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.

A.2.2 FCS_HTTPS_EXT.1 HTTPS Protocol

A.2.2.1 Family Behaviour

Components in this family define the requirements for protecting remote management sessions between the TOE and a Security Administrator. This family describes how HTTPS will be implemented. This is a new family defined for the FCS Class.

A.2.2.2 Component leveling

FCS_HTTPS_EXT HTTPS Protocol
1

Figure 5 – HTTPS Protocol family decompisition

FCS_HTTPS_EXT.1 HTTPS requires that HTTPS be implemented according to RFC 2818 and supports TLS.

A.2.2.3 Management: FCS_HTTPS_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

A.2.2.4 Audit: FCS_HTTPS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen.

A.2.2.5 FCS_HTTPS_EXT.1 HTTPS Protocol

Hierarchical to:	No other components
Dependencies:	FCS_TLSC_EXT.1 TLS Client Protocol, or FCS_TLSS_EXT.1 TLS Server 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
The TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*] if the peer certificate is deemed invalid.

A.2.3 FCS_TLSC_EXT TLS Client Protocol

A.2.3.1 Family Behaviour

The component in this family addresses the ability for a client to use TLS to protect data between the client and a server using the TLS protocol.

A.2.3.2 Component leveling

FCS_TLSC_EXT TLS Client Protocol
1
2

Figure 6 – TLS Client Protocol family decomposition

FCS_TLSC_EXT.1 TLS Client requires that the client side of TLS be implemented as specified.

FCS_TLSC_EXT.2 TLS Client requires that the client side of the TLS implementation include mutual authentication.

A.2.3.3 Management: FCS_TLSC_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

A.2.3.4 Audit: FCS_TLSC_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment
b) TLS session establishment
c) TLS session termination

A.2.3.5 FCS_TLSC_EXT.1 TLS Client Protocol

Hierarchical to:	No other components
Dependencies:	FCS_CKM. 1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment 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_RBG_EXT.1 Random Bit Generation

FCS_TLSC_EXT.1.1

The TSF shall implement [selection: *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:

- [assignment: *List of optional ciphersuites and reference to RFC in which each is defined*].

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

The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*].

FCS_TLSC_EXT.1.4

The TSF shall [selection: not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves] in the Client Hello.

A.2.4 FCS_TLSS_EXT TLS Server Protocol

A.2.4.1 Family Behaviour

The component in this family addresses the ability for a server to use TLS to protect data between a client and the server using the TLS protocol.

A.2.4.2 Component leveling

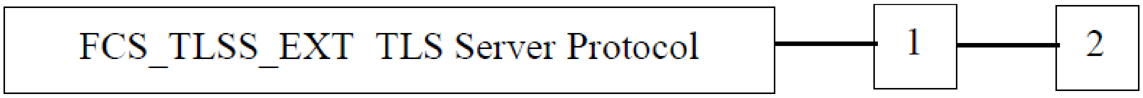

Figure 7 – TLS Server Protocol family decomposition

FCS_TLSS_EXT.1 TLS Server requires that the server side of TLS be implemented as specified.

FCS_TLSS_EXT.2: TLS Server requires the mutual authentication be included in the TLS implementation.

A.2.4.3 Management: FCS_TLSS_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

A.2.4.4 Audit: FCS_TLSS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment
b) TLS session establishment
c) TLS session termination

A.2.4.5 FCS_TLSS_EXT.1 TLS Server Protocol

Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment 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 (KeyedHash Algorithm) FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSS_EXT.1.1

The TSF shall implement [selection: *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:

- [assignment: *list of optional ciphersuites and reference to RFC in which each is defined*].

FCS_TLSS_EXT.1.2

The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [selection: *TLS 1.1, TLS 1.2, none*].

FCS_TLSS_EXT.1.3

The TSF shall [selection: *perform RSA key establishment with key size [selection: 2048 bits, 3072 bits, 4096 bits]; generate EC Diffie-Hellman parameters over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie-Hellman parameters of size [selection: 2048 bits, 3072 bits]*].

A.3 Class FIA: Identification and Authentication

Families in this class address the requirements for functions to establish and verify a claimed user identity as defined in CC Part 2.

A.3.1 Password Management (FIA_PMG_EXT)

A.3.1.1 Family Behavior

The TOE defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

A.3.1.2 Component Leveling

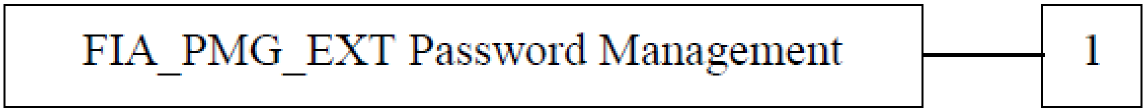

Figure 8 – Password Management family decomposition

FIA_PMG_EXT.1 Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

A.3.1.3 Management: FIA_PMG_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

A.3.1.4 Audit: FIA_PMG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen.

A.3.1.5 FIA_PMG_EXT.1 Password Management

Hierarchical to:	No other components.
Dependencies:	No dependencies.

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:* [selection: *"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: other characters]*];

b) *Minimum password length shall be* configurable to [*assignment: minimum number of characters supported by the TOE*] and [*assignment: number of characters greater than or equal to 15*].

A.3.2 User Identification and Authentication (FIA_UIA_EXT)

A.3.2.1 Family Behavior

The TSF allows certain specified actions before the non-TOE entity goes through the identification and authentication process.

A.3.2.2 Component Leveling

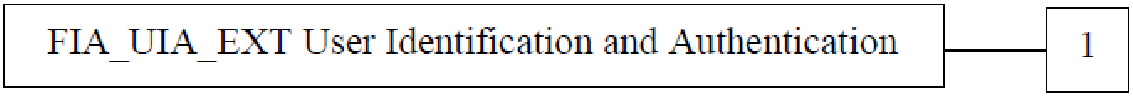

Figure 9 – User Identification and Authentication family decomposition

FIA_UIA_EXT.1 User Identification and Authentication requires administrators (including remote administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions.

A.3.2.3 Management: FIA_UIA_EXT.1

The following actions could be considered for the management functions in FMT:

a) Ability to configure the list of TOE services available before an entity is identified and authenticated.

A.3.2.4 Audit: FIA_UIA_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) All use of the identification and authentication mechanism
b) Provided user identity, origin of the attempt (e.g. IP address)

A.3.2.5 FIA_UIA_EXT.1 User identification and authentication

Hierarchical to:	No other components.
Dependencies:	FTA_TAB.1 Default TOE Access Banners.

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;
- [selection: *no other actions, [assignment: list of services, actions performed by the TSF in response to non-TOE requests]*].

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.

A.3.3 User Authentication (FIA_UAU) (FIA_UAU_EXT)

A.3.3.1 Family Behavior

Provides for a locally based administrative user authentication mechanism.

A.3.3.2 Component Leveling

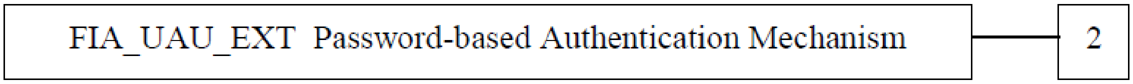

Figure 10 – Password-based Authentication Mechanism family decomposition

FIA_UAU_EXT.2 The password-based authentication mechanism provides administrative users a locally based authentication mechanism.

A.3.3.3 Management: FIA_UAU_EXT.2

The following actions could be considered for the management functions in FMT:

a) None

A.3.3.4 Audit: FIA_UAU_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All use of the authentication mechanism.

A.3.3.5 FIA_UAU_EXT.2 Password-based Authentication Mechanism

Hierarchical to:	No other components
Dependencies:	No other components

FIA_UAU_EXT.2.1

The TSF shall provide a local password-based authentication mechanism, and [selection: *[assignment: other authentication mechanism(s)], no other authentication mechanism*] to perform local administrative user authentication.

A.3.4 Authentication using X.509 certificates (Extended – FIA_X509_EXT)

A.3.4.1 Family Behavior

This family defines the behavior, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates for authentication for protocols and integrity verification, and the generation of certificate requests.

A.3.4.2 Component Leveling

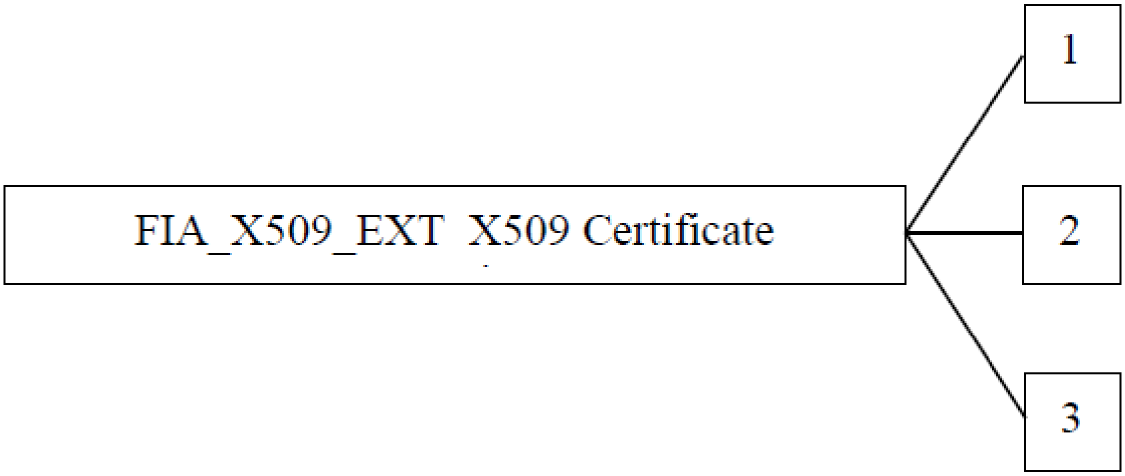

Figure 11 – X.509 Certificate family decomposition

FIA_X509_EXT.1 X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.

FIA_X509_EXT.2 X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.

FIA_X509_EXT.3 X509 Certificate Requests, require the TSF to be able to generate Certificate Request Messages and validate responses.

A.3.4.3 Management: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

The following actions could be considered for the management functions in FMT:

a) Remove imported X.509v3 certificate
b) Approve import and removal of X.509v3 certificates
c) Initiate certificate requests

A.3.4.4 Audit: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: No specific audit requirements are specified.

A.3.4.5 FIA_X509_EXT.1 X.509 Certificate Validation

Hierarchical to:	No other components
Dependencies:	FIA_X509_EXT.2 X.509 Certificate Authentication FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.1.1

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List as specified in RFC 5759 Section 5, no revocation method]*
- The TSF shall validate the extendedKeyUsage field according to the following rules: [assignment: *rules that govern contents of the extendedKeyUsage field that need to be verified*].

FIA_X509_EXT.1.2

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.

A.3.4.6 FIA_X509_EXT.2 X.509 Certificate Authentication

Hierarchical to:	No other components
Dependencies:	FIA_X509_EXT.1 X.509 Certificate Validation FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.2.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: *DTLS, HTTPS, IPsec, TLS, SSH, [assignment: other protocols], no protocols*], and [selection: *code signing for system software updates, code signing for integrity verification, [assignment: other uses], 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 [selection: *allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

A.3.4.7 FIA_X509_EXT.3 X.509 Certificate Requests

Hierarchical to:	No other components
Dependencies:	FIA_X509_EXT.1 X.509 Certificate Validation FIA_X509_EXT.2 X.509 Certificate Authentication

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

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

A.4 Class FPT: Protection of the TSF

Families in this class address the requirements for functions providing integrity and management of mechanisms that constitute the TSF and of the TSF data as defined in CC Part 2.

A.4.1 Protection of TSF Data (FPT_SKP_EXT)

A.4.1.1 Family Behavior

Components in this family address the requirements for managing and protecting TSF data, such as cryptographic keys. This is a new family modelled after the FPT_PTD Class.

A.4.1.2 Component Leveling

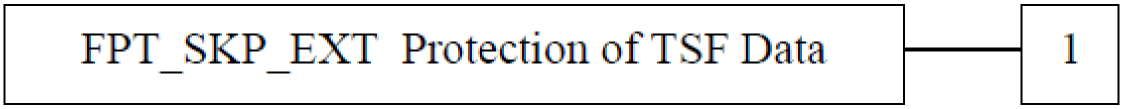

Figure 12 – Protection of TSF Data (for reading all symmetric keys) family decomposition

FPT_SKP_EXT.1 Protection of TSF Data (for reading all symmetric keys) requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

A.4.1.3 Management: FPT_SKP_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

A.4.1.4 Audit: FPT_ SKP_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen.

A.4.1.5 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

Hierarchical to:	No other components
Dependencies:	No dependencies.

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

A.4.2 Protection of Administrator Passwords (FPT_APW_EXT)

A.4.2.1 Family Behavior

Components in this family ensure that the TSF will protect plaintext credential data such as passwords from unauthorized disclosure.

A.4.2.2 Component Leveling

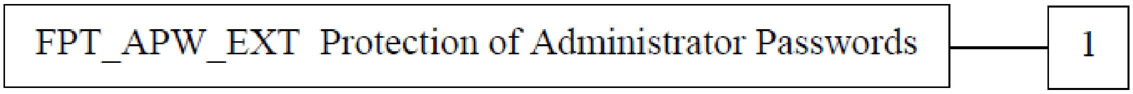

Figure 13 – Protection of Administrator Passwords family decomposition

FPT_APW_EXT.1 Protection of administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

A.4.2.3 Management: FPT_APW_EXT.1

The following actions could be considered for the management functions in FMT:

a) No management functions.

A.4.2.4 Audit: FPT_SKP_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

A.4.2.5 FPT_APW_EXT.1 Protection of Administrator Passwords

Hierarchical to:	No other components
Dependencies:	No dependencies.

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

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

A.4.3 FPT_TST_EXT.1 TSF Testing

A.4.3.1 Family Behavior

Components in this family address the requirements for self-testing the TSF for selected correct operation

A.4.3.2 Component Leveling

Figure 14 – TSF Testing family decomposition

FPT_TST _EXT.1: TSF Self-Test requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

A.4.3.3 Management: FPT_TST_EXT.1

The following actions could be considered for the management functions in FMT:

a) No management functions.

A.4.3.4 Audit: FPT_TST_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Indication that TSF self-test was completed
b) Failure of self-test

A.4.3.5 FPT_TST_EXT.1 TSF Testing

Hierarchical to: No other components
Dependencies: No other components

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]*] to demonstrate the correct operation of the TSF: [assignment: *list of self-tests run by the TSF*].

A.4.4 Trusted Update (FPT_TUD_EXT)

A.4.4.1 Family Behavior

Components in this family address the requirements for updating the TOE firmware and software.

A.4.4.2 Component Leveling

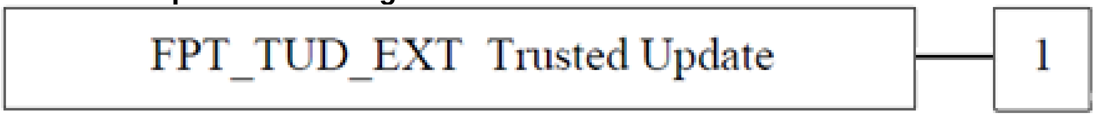

Figure 15 – Trusted Update family decomposition

FPT_TUD_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

A.4.4.3 Management: FPT_ TUD_EXT.1

The following actions could be considered for the management functions in FMT:

a) Ability to update the TOE and to verify the updates
b) Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1/SigGen) and [selection: no other functions, *[assignment:* other cryptographic functions (or other functions) used to support the update capability*]*]
c) Ability to update the TOE and to verify the updates using [selection: digital signature, published hash, no other mechanism] capability prior to installing those updates

A.4.4.4 Audit: FPT_TUD_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Initiation of the update process
b) Any failure to verify the integrity of the update

A.4.4.5 FPT_TUD_EXT.1 Trusted Update

Hierarchical to:	No other components
Dependencies:	FCS_COP.1/SigGen Cryptographic operation (for Cryptographic Signature and Verification), or FCS_COP.1/Hash Cryptographic operation (for cryptographic hashing)

FPT_TUD_EXT.1.1

The TSF shall provide [assignment: *Administrators*] the ability to query the currently executing version of the TOE firmware/software and [selection: *the most recently installed version of the TOE firmware/software; no other TOE firmware/software version*].

FPT_TUD_EXT.1.2

The TSF shall provide [assignment: *Administrators*] the ability to manually initiate updates to TOE firmware/software and [selection: *support automatic checking for updates, support automatic updates, no other update mechanism*].

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: *digital signature mechanism, published hash*] prior to installing those updates.

A.4.5 Time stamps (FPT_STM_EXT)

A.4.5.1 Family Behaviour

Components in this family extend FPT_STM requirements by describing the source of time used in timestamps.

A.4.5.2 Component leveling

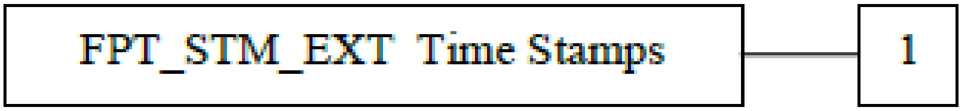

Figure 16 – Time Stamps family decomposition

FPT_STM_EXT.1 Reliable Time Stamps is hierarchic to FPT_STM.1: it requires that the TSF provide reliable time stamps for TSF and identifies the source of the time used in those timestamps.

A.4.5.3 Management: FPT_STM_EXT.1

The following actions could be considered for the management functions in FMT:

a) Management of the time
b) Administrator setting of the time

A.4.5.4 Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Discontinuous changes to the time.

A.4.5.5 FPT_STM_EXT.1 Reliable Time Stamps

Hierarchical to: No other components
Dependencies: No other components

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 [selection: *allow the Security Administrator to set the time, synchronise time with external time sources*].

A.5 Class FTA: TOE Access

Families in this class specify functional requirements for controlling the establishment of a user's session as defined in CC Part 2.

A.5.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

A.5.1.1 Family Behavior

Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

The extended FTA_SSL_EXT family is based on the FTA_SSL family.

A.5.1.2 Component Leveling

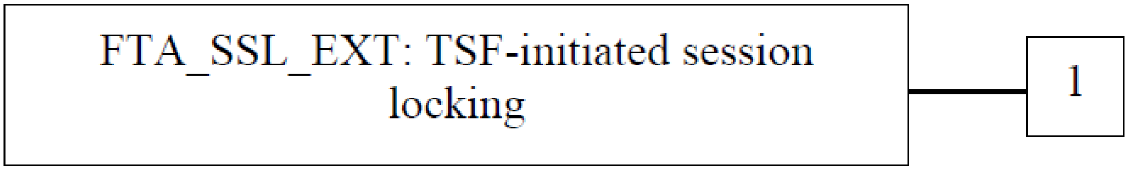

Figure 17 – TSF-initiated Session Locking family decomposition

FTA_SSL_EXT.1: TSF-initiated session locking requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

A.5.1.3 Management: FTA_SSL_EXT.1

The following actions could be considered for the management functions in FMT:

a) Specification of the time of user inactivity after which lock-out occurs for an individual user.

A.5.1.4 Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Any attempts at unlocking an interactive session.

A.5.1.5 FTA_SSL_EXT.1 TSF-initiated session locking

Hierarchical to: No other components.
Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [selection:

- *lock the session - disable any activity of the Administrator's data access/display devices other than unlocking the session, and requiring that the Administrator re-authenticate to the TSF prior to unlocking the session;*
- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

Prepared by:
Corsec Security, Inc.

13921 Park Center Road, Suite 460
Herndon, VA 20171
United States of America

Phone: +1 703 267 6050
Email: info@corsec.com
http://www.corsec.com