Bivio 6310-NC Security Target

20-5018-R-0006
Version: 0.8
Nov 25, 2020

Prepared For:
Bivio Networks, Inc.
4457 Willow Rd Suite 240
Pleasanton, CA, 94588

Prepared By:
Ekta Binwani
UL Verification Services Inc.

Table of Contents

1. Security Target (ST) Introduction

The structure of this document is defined by CC v3.1r5 Part 1 Annex A.2, “Mandatory contents of an ST”:

- Section 1 contains the ST Introduction, including the ST reference, Target of Evaluation (TOE) reference, TOE overview, and TOE description.
- Section 2 contains conformance claims to the Common Criteria (CC) version, Protection Profile (PP) and package claims, as well as rationale for these conformance claims.
- Section 3 contains the security problem definition, which includes threats, Organizational Security Policies (OSP), and assumptions that must be countered, enforced, and upheld by the TOE and its operational environment.
- Section 4 contains statements of security objectives for the TOE, and the TOE operational environment as well as rationale for these security objectives.
- Section 5 contains definitions of any extended security requirements claimed in the ST.
- Section 6 contains the security function requirements (SFR), the security assurance requirements (SAR), as well as the rationale for the claimed SFR and SAR.
- Section 7 contains the TOE summary specification, which includes the detailed specification of the IT security functions

1.1 Security Target Reference

The Security Target reference shall uniquely identify the Security Target.

ST Title:	Bivio 6310-NC Security Target
ST Version Number:	Version 0.8
ST Author(s):	Ekta Binwani
ST Publication Date:	November 25, 2020
Keywords	Network Device

1.2 Target of Evaluation Reference

The Target of Evaluation reference shall identify the Target of Evaluation.

TOE Developer	Bivio Networks, Inc. 4457 Willow Rd Suite 240 Pleasanton, CA, 94588
TOE Name:	Bivio 6310-NC
TOE Version	0.2

1.3 Target of Evaluation Overview

1.3.1 TOE Product Type

The TOE is classified as a Network Device.

1.3.2 TOE Usage

The Bivio 6310-NC device can be used to run a variety of applications for processing network data. There are many such applications, both commercial and open source. It is out of scope for this certification process to include all these applications for evaluation, so a standard application factory-installed to all Bivio 6310-NC devices as part of the base BiviOS will be provided. This application provides the following non-evaluated functionality:

- Inspects packets and will either drop them or forward them based on configuration.

- Uses the default mechanisms for packet handling and represents other packet processing applications that a customer may choose to install.

1.3.3 TOE Major Security Features Summary

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

1.3.4 TOE IT environment hardware/software/firmware requirements

The TOE requires the following hardware / infrastructure to be present:

- Syslog server conformant to RFCs 5424 (Syslog over TCP).
- A local console with a RS-232 port for use with the Bivio-provided console cable.

The TOE requires the following software to be present:

- Administrators will need an SSHv2 Client conformant to RFCs 4251, 4252, 4253, 4254, and 6668.
    - The SSHv2 client will need to be capable of supporting AES128-CBC and AES256-CBC encryption algorithms, using HMAC-SHA2-256 or HMAC-SHA2-512 integrity algorithms, and performing key exchange using Diffie-Hellman Group14-SHA1.
    - To perform public key authentication to the TOE, the SSHv2 client will need to be capable of supporting SSH-RSA.
- The TOE also provides a TLS protected server capability, which requires a TLSv1.2 client capable of negotiating one of the following ciphersuites:
    - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

  The client will connect to the TLS server, send standard input over to the TLS server, and print any data received from the server on to standard output.

  The TLS server offers a Telnet login service over TLSv1.2. Thus, the best way to utilize this service will be to use a TLS enabled Telnet client. If that is not available, a regular Telnet client can be used via a TLS proxy.

1.4 Target of Evaluation Description

The Bivio 6310-NC (Target of Evaluation, or TOE) is a network device providing highly variable network functionality. It achieves this by leveraging RHEL8.2 to provide full hardware access to the networking applications, allowing them to address the high-performance hardware devices directly. All access to the networking applications is constrained to be via the management entity, described below.

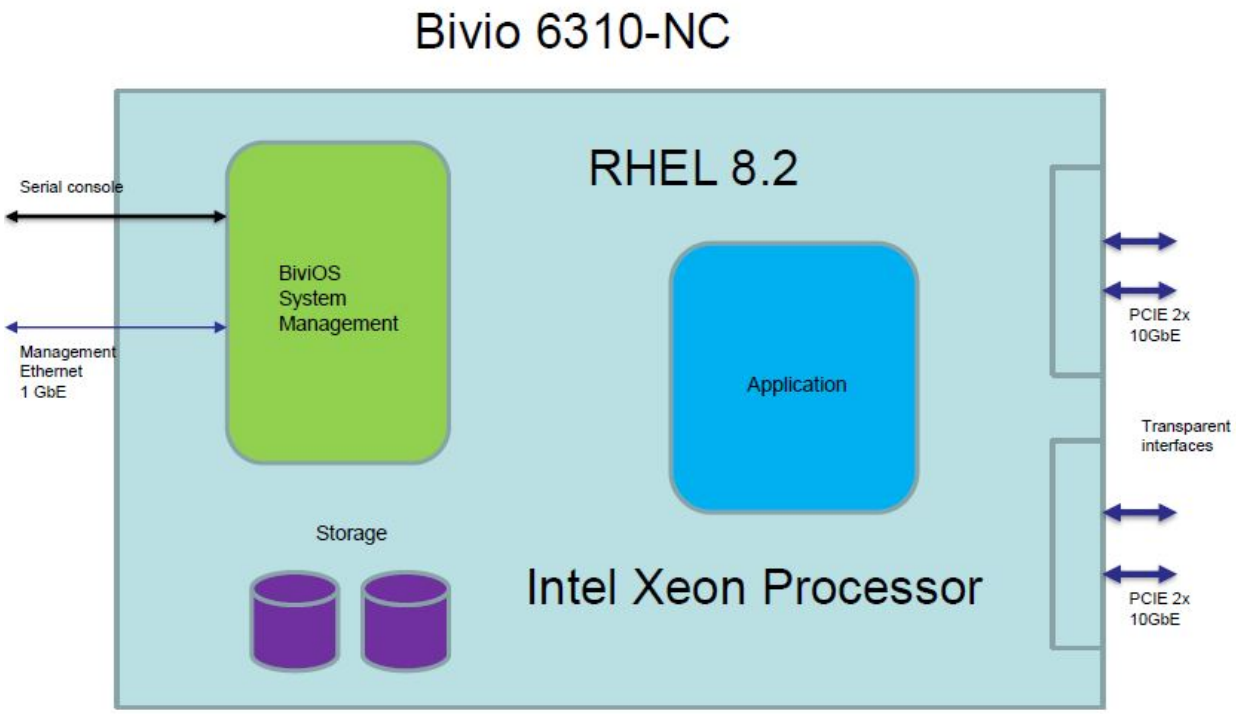

Figure 1: Bivio 6310-NC System Overview

1.4.1 TOE Physical Scope

The TOE consists of the following components:

- The management entity is BiviOS 8.5.1 over RHEL 8.2 running on an Intel Xeon processor and is the only entity accessible remotely (see Table 1 below for processor types).
- The TOE also supports an “application Entity” which is powered by the same Intel Xeon processor, running BiviOS 8.5.1 over RHEL 8.2. The only application used for the purposes of this evaluation is the standard, factory-installed application. Applications run natively, and the application entity does not employ any hypervisor.
- The application entity may not be accessed remotely, except via the management entity. Local access for maintenance/debugging purposes is provided via a serial console cable. Both the Management and Application entities directly access a shared set of processor, RAM, and storage; however, the application entity directly accesses a separate set of hardware network interfaces.
- The available application space configurations are described below.

TOEs are identified with a part number in the format:

1. B6310-NC-C(x,y)M(1,2,3,4,5)D(1,2,3,4,5,6)N(1,2,3,4)
    a. This chassis is the “standard” product chassis.
2. B6310R-NC-C(x,y)M(1,2,3)D(1,2,3,4,5,6)N(1,2,4)
    a. This chassis is a shorter, ruggedized chassis
3. PacStar 451
    a. This chassis does not have configuration options and will always use the “C04” processor specification (defined below) and no others.

The first digit ‘x’ following ‘C’ is indicative of the processor family (0 – Broadwell, 1 – Skylake, 2 – Cascade Lake), and the second digit ‘y’ (following the digit ‘x’) is selected to match Bivio’s hardware model numbering.

The naming conventions specified above reference the following hardware:

Table 1: Bivio 6310-NC Naming Convention	
Part Number	Processor
Options with C11	Dual Intel Xeon Gold 6148, 2.4 GHz w/ 27Mb Cache
Options with C13	Dual Intel Xeon Silver 4110, 2.1 GHz w/ 11Mb Cache
Options with C15	Dual Intel Xeon Gold 6138, 2.0 GHz w/27Mb Cache
Options with C21	Dual Intel Xeon Silver 4215, 2.5Ghz with 11Mb cache
Options with C22	Dual Intel Xeon Silver 4214, 2.5Ghz with 11Mb cache
Options with C23	Dual Intel Xeon Silver 4208, 2.1Ghz with 11Mb cache
Options with C24	Dual Intel Xeon Gold 5222, 3.8Ghz with 16.5 Mb cache
Options with C25	Dual Intel Xeon Gold 6242, 2.8Ghz with 22Mb cache
Options with C26	Dual Intel Xeon Gold 6252, 2.5Ghz with 35.75Mb cache
Options with C04	Intel Xeon D 1541, 2.1Ghz with 12MB cache
Part Number	Installed RAM
Options with M1	256GB DDR4-2666 memory
Options with M2	512GB DDR4-2666 memory
Options with M3	384GB DDR4-2666 memory
Options with M4	768GB DDR4-2666 memory
Options with M5	1536GB DDR4-2666 memory
Part Number	Installed Storage
Options with D1	2x 1TB SSD storage
Options with D2	2x 2TB SSD storage
Options with D3	4x 2TB SSD storage
Options with D4	8x 2TB SSD storage
Options with D5	4x 3.8TB SSD storage
Options with D6	8x 3.8TB SSD storage
Part Number	Installed NIC Interfaces
Options with N1	2x 10GbE Fiber interfaces and 4x 1GbE Copper interfaces
Options with N2	4x 10GbE Fiber interfaces and 4x 1GbE Copper interfaces
Options with N3	6x 10GbE Fiber interfaces and 2x 1GbE Copper interfaces
Options with N4	4x 10GbE Fiber interfaces and 2x 1GbE Copper interfaces

All “M”, “D”, and “N” options are configuration options which do not affect validation, but are part of the model number.
Running:
BiviOS 8.5.1 V: Version 8.5.1 (Build 202006181129) V: Version 8.5.1-104-bv (Patch 202009191230) V: Version 8.5.1-103-rh (Patch 202008311617)

AES-NI technology is enabled for all the listed CPUs.

The guidance documentation that is part of the TOE is listed in Section 9 “References” within Table 9: TOE Guidance Documentation

- The TOE hardware described is delivered to the customer via courier.

- The Software is pre-installed on the hardware prior to delivery. It is also available as a binary ISO file that can be requested by the customer.
- The guidance documentation that is part of the TOE is listed in Section 9 “References” within Table 9: TOE Guidance Documentation. This documentation is included, as a PDF file on optical media, with the shipment of the TOE hardware.

1.4.2 TOE Logical Scope

The logical boundary of the TOE includes those security functions implemented exclusively by the TOE. These security functions are listed in Section 1.3.3 above and are further described in the following subsections. A more detailed description of the implementation of these security functions are provided in Section 7 “TOE Summary Specification”.

1.4.2.1 Audit

- The TOE will audit all events and information defined in Table 4: Auditable Events.
- The TOE will also include the identity of the user that caused the event (if applicable), date and time of the event, type of event, and the outcome of the event.
- The TOE protects storage of audit information from unauthorized deletion.
- The TOE prevents unauthorized modifications to the stored audit records.
- The TOE can transmit audit data to an external IT entity using SSH protocol.

1.4.2.2 Cryptographic Operations

The TSF performs the following cryptographic operations.
For TLS:

- AES-128 in CBC mode for data ciphering, using SHA-1 hashing and RSA key exchange.
- AES-256 in GCM mode for data ciphering, using SHA-384 hashing and ECDHE key exchange.
- HMAC-SHA2-384 for keyed hash

For SSH:

- AES-128 or AES-256 in CBC mode, HMAC-SHA2-256 or HMAC-SHA2-512 hashing and DH key exchange.
- Public key authentication via SSH-RSA, RSA-SHA2-256 and RSA-SHA2-512 using HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512 hashing algorithms.

The TOE supports NTP v4 (RFC 5905). In RHEL 8.2, the NTP protocol is implemented by the chronyd daemon, which is part of the chrony package. For NTP, the TOE uses the Symmetric key Method to ensure authenticity and integrity. The TOE supports SHA1, SHA 512 and SHA256 as the MACs.

RHEL 8.2 OpenSSL v1.1.1c-15-B1, provides Random bit generation crypto module, which utilizes CTR_DRBG as defined by NIST SP 800-90A. This is not configurable, and there are no other cryptographic engines provided in the TOE.

- To support SSH for trusted path and trusted channel, the TOE cryptographic module implements RSA key generation with key sizes of 2048-bits and finite-field cryptography with modulus sizes of 2048 bits (Diffie-Hellman Group14).
- To support TLS, the TOE cryptographic module implements Elliptic-Curve key generation over NIST curve secp256r1 and RSA key generation using 2048-bit keys.

The TOE supports Trusted Update by allowing the administrator to download update files from Bivio. Any software installed on the TOE will become active immediately and is authenticated using a published hash. Trusted update uses SHA-256 hash function in its algorithm.

The TSF zeroizes all plaintext secret and private cryptographic keys and CSPs once they are no longer required.

1.4.2.3 Identification and Authentication

- The TSF supports passwords consisting of alphanumeric and special characters. The TSF also allows administrators to set a minimum password length and support passwords with 9 characters or more.
- The TSF requires all administrative users to authenticate before allowing the user to perform any actions other than:
    - Display the warning banner in accordance with FTA_TAB.1;
    - Responding to ICMP echo requests
    - Responding to ARP requests with ARP replies
    - Establishing TLS connection on TCP port 27777
    - Automated generation of cryptographic keys

1.4.2.4 Security Management

- The TSF stores and protects the following data:
    - Syslog data, user account data, and local authentication data (such as administrator passwords).
    - Cryptographic keys including pre-shared keys, symmetric keys, and private keys.
- There is one class of user on the TOE: The Admin user
    - The Admin user has full control over the TOE.
- Management of the TSF:
    - The administrator can perform manual updates, determine the behavior of or modify the behavior of the handling of audit data, modify the behavior of the TSF, enable or disable services offered by the TOE, determine the behavior of or modify the behavior of audit functionality when local audit storage is full, manage TSF data, modify, delete, generate or import cryptographic keys, configure the access banner, and configure the session inactivity timeout period.
    - The administrator may perform these functions locally or remotely using the trusted path provided by SSH and defined in FTP_TRP.1.

1.4.2.5 Protection of the TSF

- The TSF protects TSF data from disclosure when the data is transmitted between different parts of the TOE.
- The TSF prevents the reading of secret and private keys.
- The TOE provides reliable time stamps for itself.
- The TOE runs a suite of self-tests during the initial start-up (upon power on) to demonstrate the correct operation of the TSF.
- The TOE provides a means to verify firmware/software updates to the TOE using a published hash prior to installing those updates.

1.4.2.6 TOE Access

- The TOE, for local interactive sessions, will terminate the session after an Authorized Administrator-specified period of session inactivity.
- The TOE terminates a remote interactive session after an Authorized Administrator-configurable period of session inactivity.
- The TOE allows Administrator-initiated termination of the Administrator's own interactive session.
- Before establishing an administrative user session, the TOE is capable of displaying an Authorized Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE.

1.4.2.7 Trusted Path/Channels

- The TOE uses SSH to provide a trusted communication channel between itself and all authorized IT entities 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.
- The TOE permits the TSF, or the authorized IT entities to initiate communication via the trusted channel.
- The TOE permits remote administrators to initiate communication via the trusted path.
- The TOE requires the use of the trusted path for initial administrator authentication and all remote administration actions.

1.5 Notation, formatting, and conventions

The notation, formatting, and conventions used in this Security Target are defined below; these styles and clarifying information conventions were developed to aid the reader.
Where necessary, the ST author has added application notes to provide the reader with additional details to aid understanding; they are italicized and usually appear following the element needing clarification.
The notation conventions that refer to iterations, assignments, selections, and refinements made in this Security Target are in reference to SARs and SFRs taken directly from CC Part 2 and Part 3 as well as any SFRs and SARs taken from a Protection Profile.
The notation used in those PP to indicate assignments, selections, and refinements of SARs and SFRs taken from CC Part 2 and Part 3 is not carried forward into this document. Additionally, obvious errors in the PP are corrected and noted as such.
The CC permits four component operations (assignment, iteration, refinement, and selection) to be performed on requirement components. These operations are defined in Common Criteria, Part 1; Section 8.1, "Operations" as:

- Iteration: allows a component to be used more than once with varying operations;
- Assignment: allows the specification of parameters;
- Selection: allows the specification of one or more items from a list; and
- Refinement: allows the addition of details.

Iterations performed by the ST author are indicated by a number in parenthesis following the requirement number, e.g., FIA_UAU.1.1(1); the iterated requirement titles are similarly indicated, e.g., FIA_UAU.1(1). Iterations performed by the PP author are indicated by a slash followed by a short description, e.g. FCS_COP.1/Hash.
Assignments are identified with **bold text.**
Selections are identified with <u>underlined text</u>**.**
Refinements that add text use ***bold and italicized text*** to identify the added text***.*** Refinements that performs a deletion, identifies the deleted text with ~~***strikeout, bold, and italicized text***~~.

2. Conformance Claims

2.1 Common Criteria Conformance Claims

This Security Target and TOE are conformant to the Common Criteria Version 3.1 Release 5, CC Part 2 extended [C2], and CC Part 3 conformant [C3].

2.2 Conformance to Protection Profiles

This Security Target claims exact compliance to the collaborative Protection Profile for Network Devices, Version 2.2e, dated March 23, 2020 [cPP]. This Protection Profile will be referred to as cPP or PP for convenience throughout this Security Target.
The TOE conforms with the following Technical Decisions:

Table 2: Technical Decisions

TD	TD Title	TOE Applicability
0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes
0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	Yes
0536	NIT Technical Decision for Update Verification Inconsistency	Yes
0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	No
0538	NIT Technical Decision for Outdated link to allowed-with list	Yes
0546	NIT Technical Decision for DTLS – clarification of Application Note 63	No
0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes
0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	Yes
0556	NIT Technical Decision for RFC 5077 question	Yes

2.3 Conformance to Security Packages

This Security Target does not claim conformance to any security function requirements or security assurance requirements packages, neither as package-conformant or package-augmented.

2.4 Conformance Claims Rationale

To demonstrate that exact conformance is met, this rationale shows all threats are addressed, all OSP are satisfied, no additional assumptions are made, all objectives have been addressed, and all SFRs and SARs have been instantiated.
The following address the completeness of the threats, OSP, and objectives, limitations on the assumptions, and instantiation of the SFRs and SARs:

- Threats
  - All threats defined in the cPP;
  - No additional threats have been defined in this ST.
- Organizational Security Policies
  - All OSP defined in the cPP are carried forward to this ST;
  - No additional OSPs have been defined in this ST.
- Assumptions

    - All assumptions defined in the cPP for a standalone TOE are carried forward to this ST;
    - No additional assumptions for the operational environment have been defined in this ST.
- Objectives
    - All objectives defined in the cPP for a standalone TOE are carried forward to this ST. Optional and selection based SFRs defined in the cPP are carried forward to this Security Target as required by the cPP.
- All mandatory SFRs and SARs defined in the cPP are carried forward to this Security Target.

Rationale presented in the body of this ST shows all assumptions on the operational environment have been upheld, all the OSP are enforced, all defined objectives have been met and these objectives counter the defined threats.
Additionally, all SFRs and SARs defined in the cPP have been properly instantiated in this Security Target; therefore, this ST shows exact compliance to the cPP.

3. Security Problem Definition

3.1 Threats

The following section defines the security threats for the TOE, characterized by a threat agent, an asset, and an adverse action of that threat agent on that asset. These threats are taken directly from the PP unchanged.

T.UNAUTHORIZED_ADMINISTRATOR_ACCESS
Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.

T.WEAK_CRYPTOGRAPHY
Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.

T.UNTRUSTED_COMMUNICATION_CHANNELS
Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.

T.WEAK_AUTHENTICATION_ENDPOINTS
Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.

T.UPDATE_COMPROMISE
Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.

T.UNDETECTED_ACTIVITY
Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.

T.SECURITY_FUNCTIONALITY_COMPROMISE
Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.

T.PASSWORD_CRACKING
Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other Network Devices.

T.SECURITY_FUNCTIONALITY_FAILURE
An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

3.2 Organizational Security Policies

The following section defines the organizational security policies which are a set of rules, practices, and procedures imposed by an organization to address its security needs. These threats are taken directly from the PP unchanged.

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.

3.3 Assumptions

This section describes the assumptions on the operational environment in which the TOE is intended to be used. It includes information about the physical, personnel, and connectivity aspects of the environment. The operational environment must be managed in accordance with the provided guidance documentation. The following table defines specific conditions that are assumed to exist in an environment where the TOE is deployed. These assumptions are taken directly from the PP unchanged.

A.PHYSICAL_PROTECTION
The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.

A.LIMITED_FUNCTIONALITY
The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).
In the case of vNDs, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of the distributed TOE run inside more than one virtual machine (VM) on a single VS. There are no other guest VMs on the physical platform providing non-Network Device functionality.

A.NO_THRU_TRAFFIC_PROTECTION
A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND

cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).

A.TRUSTED_ADMINISTRATOR

The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.

For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).

A.REGULAR_UPDATES

The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

A.ADMIN_CREDENTIALS_SECURE

The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.

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.

4. Security Objectives

4.1 Security Objectives for the Operational Environment

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

5. Extended Components Definition

As stated in Section 2, this Security Target claims exact conformance to the referenced cPP and modules. As such, the extended components definition is contained in the cPP and modules claimed. In this case the cPP is Part 3 conformant and so there are no extended SARs defined.

6. Security Requirements

This section describes the security functional and assurance requirements for the TOE.

6.1 Security Functional Requirements

This section describes the functional requirements for the TOE. The security functional requirement components in this security target are CC Part 2 conformant or CC Part 2 extended as defined in Section 2, Conformance Claims. Operations that were performed in the cPP are not signified in this section. Operations performed by the ST are denoted according to the formatting conventions in Section 1.5.

Table 3: Security Functional Requirements

SFR	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG.1	Protected audit trail storage (Optional)
FAU_STG_EXT.1	Protected Audit Event Storage
FAU_STG_EXT.3 /LocSpace	Action in case of possible audit data loss (Optional)
FCS_CKM.1	Cryptographic Key Generation (Refinement)
FCS_CKM.2	Cryptographic Key Establishment (Refinement)
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_NTP_EXT.1	NTP Protocol (Selection-based)
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHC_EXT.1	SSH Client (Selection-based)
FCS_SSHS_EXT.1	SSH Server (Selection-based)
FCS_TLSS_EXT.1	TLS Server Protocol (Selection-based)
FIA_AFL.1	Authentication Failure Management (Refinement)
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation (Selection-based)
FIA_X509_EXT.2	X.509 Certificate Authentication (Selection-based)
FIA_X509_EXT.3	X.509 Certificate Requests (Selection-based)
FMT_MOF.1 /Functions	Management of security functions behavior (Selection-based)

Table 3: Security Functional Requirements	
SFR	Description
FMT_MOF.1 /ManualUpdate	Management of security functions behavior
FMT_MOF.1 /Services	Management of security functions behavior (Selection-based)
FMT_MTD.1 /CoreData	Management of TSF Data
FMT_MTD.1 /CryptoKeys	Management of TSF data (Selection-based)
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 (Extended)
FPT_TUD_EXT.1	Trusted Update
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination (Refinement)
FTA_SSL.4	User-initiated Termination (Refinement)
FTA_TAB.1	Default TOE Access Banners (Refinement)
FTP_ITC.1	Inter-TSF trusted channel (Refinement)
FTP_TRP.1/Admin	Trusted Path (Refinement)

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:
   - Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).
   - Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
   - Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
   - Resetting passwords (name of related user account shall be logged).
   - **All entered commands.**
d) Specifically defined auditable events listed in ***Table 4***.

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of ***Table 4***.

Table 4: Auditable Events		
SFR	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1 (optional)	None.	None.
FAU_STG_EXT .1	None.	None.
FAU_STG_EXT .3/LocSpace (optional)	Low storage space for audit events.	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_NTP_EXT. 1 (selection-based)	Configuration of a new time server Removal of configured time serve	Identity if new/removed time server
FCS_RBG_EXT .1	None.	None.
FCS_SSHC_EXT.1 (selection-based)	Failure to establish an SSH session	Reason for failure
FCS_SSHS_EXT.1 (selection-based)	Failure to establish an SSH session	Reason for failure
FCS_TLSS_EXT.1 (selection-based)	Failure to establish a TLS Session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT. 1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT. 2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).

Table 4: Auditable Events		
SFR	Auditable Events	Additional Audit Record Contents
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev (selection-based)	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2 (selection-based)	None	None
FIA_X509_EXT.3 (selection-based)	None.	None.
FMT_MOF.1/Functions (selection-based)	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Services (Selection-based)	None.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys (selection-based)	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_SKP_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).

Table 4: Auditable Events		
SFR	Auditable Events	Additional Audit Record Contents
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.

6.1.1.2 FAU_GEN.2 User Identity Association

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

6.1.1.3 FAU_STG.1 Protected Audit Trail Storage (Optional)

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

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

6.1.1.4 FAU_STG_EXT.1 Protected Audit Event Storage

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

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

- The TOE shall consist of a single standalone component that stores audit data locally.

FAU_STG_EXT.1.3
The TSF shall **not store the audit data locally and send the audit data to an external IT entity** when the local storage space for audit data is full.

6.1.1.5 FAU_STG_EXT.3/LocSpace Action in Case of Possible Audit Data Loss (Optional)

FAU_STG_EXT.3.1/LocSpace

The TSF shall generate a warning to inform the Administrator before the audit trail exceeds the local audit trail storage capacity.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation (Refinement)

FCS_CKM.1.1

The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm:

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

6.1.2.2 FCS_CKM.2 Cryptographic Key Establishment (Refinement)

FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1;
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;
- FFC Schemes using ”safe-prime” groups that meet the following: ‘NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and RFC 3526.

6.1.2.3 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1

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

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

that meets the following: No Standard.

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

FCS_COP.1.1/DataEncryption

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

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

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

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) **2048 bits**,

that meet the following:

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,

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

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

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

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

6.1.2.8 FCS_NTP_EXT.1 NTP Protocol (Selection-based)

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

FCS_NTP_EXT.1.2
The TSF shall update its system time using

- Authentication using: SHA1, SHA256, SHA512 as the message digest algorithm(s);

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

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

6.1.2.9 FCS_RBG_EXT.1 Random Bit Generation

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

FCS_RBG_EXT.1.2

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

6.1.2.10 FCS_SSHC_EXT.1 SSH Client Protocol (Selection-based)

FCS_SSHC_EXT.1.1
The TSF shall implement the SSH protocol in accordance with RFCs 4251, 4252, 4253, 4254, 6668.

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

FCS_SSHC_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than **262144 (256*1024)** bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.4
The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-cbc, aes256-cbc.

FCS_SSHC_EXT.1.5
The TSF shall ensure that the SSH public-key based authentication implementation uses ssh-rsa as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHC_EXT.1.6
The TSF shall ensure that the SSH transport implementation uses hmac-sha2-256, hmac-sha2-512 as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7
The TSF shall ensure that diffie-hellman-group14-sha1 and no other methods are the only allowed key exchange methods used for the SSH protocol.

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

FCS_SSHC_EXT.1.9
The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key and no other methods as described in RFC 4251 section 4.1.

6.1.2.11 FCS_SSHS_EXT.1 SSH Server Protocol (Selection-based)

FCS_SSHS_EXT.1.1
The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, 6668, 8332.

FCS_SSHS_EXT.1.2

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

FCS_SSHS_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than **262144 (256*1024)** bytes in an SSH transport connection are dropped.

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

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

FCS_SSHS_EXT.1.6
The TSF shall ensure that the SSH transport implementation uses hmac-sha2-256, hmac-sha2-512 as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7
The TSF shall ensure that diffie-hellman-group14-sha1 and no other methods are the only allowed key exchange methods used for the SSH protocol.

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

6.1.2.12 FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication (Selection-based)

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_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

and no other ciphersuites.

FCS_TLSS_EXT.1.2
The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and TLS 1.1.

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

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

6.1.3 Identification and Authentication (FIA)

6.1.3.1 FIA_AFL.1 Authentication Failure Management (Refinement)

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within **the range 1 to 16348** unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until **the action to unlock the session by logging in over a local serial console** is taken by an Administrator; prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed.

6.1.3.2 FIA_PMG_EXT.1 Password Management

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", **all other printable characters**;
b) Minimum password length shall be configurable to between **9** and **128** characters.

6.1.3.3 FIA_UIA_EXT.1 User Identification and Authentication

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

- Display the warning banner in accordance with FTA_TAB.1;
- **Respond to ICMP Echo Request, respond to ARP requests with ARP replies, establish TLS connection on TCP port 27777, automated generation of cryptographic keys**

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.

6.1.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

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

6.1.3.5 FIA_UAU.7 Protected Authentication Feedback

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

6.1.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation (Selection-based)

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

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

6.1.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication (Selection-based)

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 not accept the certificate.

6.1.3.8 FIA_X509_EXT.3 X.509 Certificate Requests (Selection-based)

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

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

6.1.4 Security Management (FMT)

6.1.4.1 FMT_MOF.1/Functions Management of Security Functions Behavior (Selection-based)

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

6.1.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

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

6.1.4.3 FMT_MOF.1/Services Management of Security Functions Behavior (Selection-based)

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

6.1.4.4 FMT_MTD.1/CoreData Management of TSF Data

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

6.1.4.5 FMT_MTD.1/CryptoKeys Management of TSF data (Selection-based)

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

6.1.4.6 FMT_SMF.1 Specification of Management Functions

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

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using hash comparison capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to start and stop services
- Ability to modify the behaviour of the transmission of audit data to an external IT entity;
- Ability to manage the cryptographic keys;
- Ability to configure the cryptographic functionality;
- Ability to configure thresholds for SSH rekeying;
- Ability to re-enable an Administrator account;
- Ability to set the time which is used for time-stamps;
- Ability to configure NTP
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
- Ability to import X.509v3 certificates to the TOE's trust store;
- No other capabilities.

6.1.4.7 FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1
The TSF shall maintain the roles:

- Security Administrator.

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

FMT_SMR.2.3
The TSF shall ensure that the conditions

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

are satisfied.

6.1.5 Protection of the TSF (FPT)

6.1.5.1 FPT_APW_EXT.1 Protection of Administrator Passwords

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

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

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

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

6.1.5.3 FPT_STM.EXT.1 Reliable Time Stamps

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

FPT_STM_EXT.1.2
The TSF shall allow the Security Administrator to set the time, synchronise time with an NTP Server.

6.1.5.4 FPT_TST_EXT.1 TSF Testing (Extended)

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests during initial start-up (on power on), periodically during normal operation, at the request of the authorized user to demonstrate the correct operation of the TSF: **memory, RDSEED, software integrity, cryptographic tests.**

6.1.5.5 FPT_TUD_EXT.1 Trusted Update

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

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

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

6.1.6 TOE Access (FTA)

6.1.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

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

- terminate the session

after a Security Administrator-specified time period of inactivity.

6.1.6.2 FTA_SSL.3 TSF-initiated Termination (Refinement)

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

6.1.6.3 FTA_SSL.4 User-initiated Termination (Refinement)

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

6.1.6.4 FTA_TAB.1 Default TOE Access Banners (Refinement)

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.

6.1.7 Trusted path/channels (FTP)

6.1.7.1 FTP_ITC.1 Inter-TSF Trusted Channel (Refinement)

FTP_ITC.1.1
The TSF shall be capable of using SSH to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, no other capabilities that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

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

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

6.1.7.2 FTP_TRP.1/Admin Trusted Path (Refinement)

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

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

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

6.2 Security Assurance Requirements

This Security Target is conformant with the assurance requirements specified in the cPP.

Table 5: Assurance Requirements	
Assurance Class	Assurance Component
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 coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

7. TOE Summary Specification

This section provides evaluators and potential consumers of the TOE with a high-level description of each SFR, thereby enabling them to gain a general understanding of how the TOE is implemented. These descriptions are intentionally not overly detailed, thereby disclosing no proprietary information. These sections refer to SFRs defined in Section 6, Security Requirements.

The TOE consists of the following Security Functions:

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

7.1 Security Audit

7.1.1 Audit Data Generation

The TOE creates and stores audit records for the following events:

Start up and shutdown of the audit function (as startup and shutdown messages for the OS, since logging may not be started or stopped independently of the TOE startup and shutdown).

All administrative actions, including:

- Administrative login and logout, including username
- Security related 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)
- Starting and stopping services
- All commands entered during administrative sessions

All events specified in Table 4: Auditable Events.

Auditing is done using the Linux audit server (auditd). The audit server is configured to store the audit logs locally and transmit them to the administrator-configured audit server via SSH. Local audit logs are stored in /var/log/audit/audit.log in the underlying file system. Only an authorized administrator can read log files, modify or delete log files, or archive log files through the CLI, and such actions require being first authenticated as an authorized administrator using the trusted path provided in FTP_TRP.1. The TOE only defines one user type, “administrator”.

For each audit event, the TSF associates the audit event with the identity of the user that caused the event. Each audit log contains a date-and-time stamp of the event, the type of event, subject identity, and outcome (success or failure) of the event. Additional information, if available, is also logged.

Audit logs are created for generating/import of, changing, or deleting of cryptographic keys. The following scenarios create an audit log:

- Deletion of cryptographic keys using zeroize-keyfile command
- Deletion of cryptographic keys using rm command
- Importing a signed signature
- Importing a CA certificate bundle
- Generating a cryptographic key and certificate request for TLS login

- Generating self-signed certificate
- Regenerating SSH host keys (public and private) using the command ssh-host-keygen
- Generating public and private keys for the SSH tunnel client when ssh-tunnel-keygen utility is used

FAU_GEN.1
FAU_GEN.2

7.1.2 Audit Storage

The TOE stores audit records locally as well as transmits them to an external, administrator-configurable audit server. Such audit records are unable to be viewed, modified, or deleted except by an authorized administrator.
FAU_STG.1 (Optional)
The TOE transmits audit data to an external audit server via SSH, provided by FTP_ITC.1. The transmission of data to an external server is real-time.
How the TOE stores data is configurable by the administrator. The whole partition on the management entity's onboard storage (effectively about 50GB) is available for storing audit logs. However, the configuration should prevent the partition being filled. The administrator can set up the configuration by editing the file /etc/audit/auditd.conf.
All stored audit logs and archived files are protected against unauthorized deletion, modification, or viewing. To view these files, a user must be authenticated as an authorized administrator via the trusted path provided by SSH (FTP_TRP.1).
FAU_STG_EXT.1

When the disk space left in the audit log partition falls below the "space_left" parameter, an audit record is created as a warning to the Administrator. This warning message will be posted in the local syslog (/var/log/messages) and will also be forwarded on to the remote audit log server, but it will not be stored in the local audit log file. Also, while this condition persists, all new audit logs will be sent to the local syslog (/var/log/messages) and also forwarded on to the remote audit log server, but they will not be stored in the local audit log file.
FAU_STG.3/LocSpace (Optional)

7.2 Cryptographic Support

The TOE utilizes the following CAVP validated algorithms in the Certificate Validation Number C1935 in the evaluated configuration:

Table 6: CAVP Certificates

Function	SFR	Comments
AES - Encryption / Decryption	FCS_COP.1(1)	The TOE performs AES in the CBC, GCM or GMAC mode with key sizes 128 or 256 bits.
ECDH	FCS_CKM.2	ECDH is used in TLS EC session key establishment.
RSA	FCS_COP.1(2) FCS_CKM.1	The TOE performs RSA with 2048-bit moduli.
HMAC	FCS_COP.1(4)	The TOE performs HMAC using the following: HMAC-SHA1, with a key size of 160 bits and an output digest size of 160 bits. HMAC-SHA2-256 with a key size of 256 bits and an output digest size of 256 bits. HMAC-SHA2-384 with a key size of 384 bits and an output digest size of 384 bits. HMAC-SHA2-512 with a key size of 512 bits and an output digest size of 512 bits.

Cryptographic Hashing	FCS_COP.1(3)	The TOE performs cryptographic hashing as follows: SHA1 - used in SSH, TLS HMAC. SHA-256 - used in SSH HMAC and TLS HMAC. SHA-384 - used in TLS and SSH HMAC. SHA-512 – used in TLS and SSH HMAC Appropriate hash algorithms are also used for digital signature verification, for NTP message authentication and the key-derivation functions of SSH and TLS.
DRBG	FCS_RBG_EXT.1	Random bit generation

The TOE uses OpenSSH 8.0p1-4 to provide all SSH functions, including the trusted path and trusted channel.

7.2.1 Cryptographic Key Generation

To support SSH for trusted path and trusted channel, the TOE cryptographic module implements RSA key generation with key sizes of 2048-bits and finite-field cryptography key generation with modulus sizes of 2048 bits (Diffie-Hellman Group14). To support TLS, the TOE cryptographic module implements Elliptic-Curve key generation over NIST curve secp256r1 and RSA key generation using 2048-bit keys.

FCS_CKM.1

The TOE implements Elliptic Curve-based key establishment. Diffie-Hellman key establishment with RSA utilizing key sizes of 2048 bits is used in the SSH implementation, while elliptic curve-based key establishment with ECDH is used to establish the TLS tunnel. For Diffie-Hellman, the TOE meets RFC 3526 Section 3 by copying the parameters from the RFC and using them directly in prime generation.

The RSA-based key establishment used by the TOE is RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. The RSA signature is being performed using PKCS #1 v2.1 Signature Schemes RSASSA-PSS or RSASSA-PKCS1v1_5.

Scheme	SFR	Service
Diffie-Hellman Group-14 SHA-1	FCS_SSHS_EXT.1 FCS_SSHC_EXT.1	1. Administration via SSH v2 2. Audit Server (uses SSH tunnel)
RSA ECDHE_RSA	FCS_TLSS_EXT.1	Administration via TLS-v1.2

FCS_CKM.2

Plaintext keys are temporary session keys for TLS or SSH, or persistent keys currently being used to perform cryptographic operations. Plaintext temporary session keys are generated on-the-fly as SSH and TLS sessions are created and are not stored outside of RAM. Zeroization of keys in RAM is accomplished by overwriting with zeroes, followed by a read-verify. If the read-verification fails, the process is repeated.

Temporary keys are stored in RAM (volatile memory). These are zeroized when the associated session is ended.

Key generation for TLS server can be performed by tls-server-csr-gen command. This will create an unencrypted private key(stunnel.key), and a certificate request (req.pem). The key file can now be used as the TLS server's private key, and the request file can be shipped to a CA to be signed. The signed certificate can then be imported as the TLS server's public key certificate.

Persistent keys are stored on SSDs. For SSDs, the destruction of keys is executed by a block erase, followed by a read-verify. If this read verification fails, the process is repeated again. The only private keys stored on SSDs are those used by sshd, the ssh tunnel client, and the TLS server. Zeroization of such keys is performed by executing the “zeroize-keyfile” command by the administrator when such keys are no longer required. The private key must be zeroized before it is deleted. Public/private key pair is created for the tunnel client using “ssh-tunnel-keygen”.

The ssh-host-keygen utility generates RSA keys for sshd, namely, the key pair ssh_host_rsa_key and ssh_host_rsa_key.pub. If the keys are generated on a remote system only the public key needs to be imported to Bivio 6310-NC using a secure copy utility on the remote system. The host keys used by the sshd program are stored in /etc/ssh. The key type supported is RSA. The key used for the SSH secure channel (for transporting audit data to a remote server) is stored in /opt/ssh-tunnel-client. The key type supported is RSA. The key used by the TLS server is stored under /opt/TLS-server. The key type supported is RSA.

For plaintext keys in volatile storage, after the key is overwritten by zeros, references to this key are destroyed by setting any pointers to this key in memory to the NULL pointer value. Thus, there will not be any data structure in memory that will be pointing to this key after it is destroyed. The garbage collection request is just the returning of the memory used by this key to the free memory pool.

FCS_CKM.4

7.2.2 Cryptographic Operations

The TOE performs encryption and decryption in accordance with AES in CBC and GCM modes, using key sizes of 128 or 256 bits. These options are enforced in the OpenSSL configuration files and are configured at the factory.

The TOE uses RSA digital signatures for all cryptographic functions where a digital signature is required. This is enforced in the OpenSSL configuration files, which are configured at the factory. The RSA key length in 2048 bits.

The TOE performs SHA-1, SHA-256, SHA-384, and SHA-512 hashing. This is enforced in the OpenSSL configuration files, which are configured at the factory. These hashes are used in digital signature verification; for NTP message authentication; the keyed-hash message authentication code in SSH as both client and server, and in TLS; to perform password hashing; and to support file integrity checking. The TOE generates hashes with message digest sizes of 160, 256, 384, or 512 bits.
The TOE performs keyed-hash message authentication using HMAC-SHA1 (public key authentication for SSH-RSA), HMAC-SHA-256, HMAC-SHA-384, or HMAC-SHA-512 and using cryptographic key sizes of 160, 256, 384, 512. HMAC message digest sizes of 160, 256, 384, or 512 bits are used, with block sizes of 64 bytes for SHA-1 and SHA-256, and 128 bytes for SHA-384 and SHA-512, per RFC2104 and RFC4868. All of these options are administrator-configurable where not mandated by a specific protocol or configuration. For instance, certain SSH integrity algorithms are specified by the SSH SFRs. This is enforced in the OpenSSL configuration files, which are configured at the factory.

FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash

7.2.3 NTP Protocol (Selection-based)

The NTP supported by the TOE is NTP v4 (RFC 5905). In RHEL 8.2, the NTP protocol is implemented by the chronyd daemon, which is part of the chrony package.
The TOE uses the Symmetric Key method to ensure the timestamp it receives from an NTP timeserver is from an authenticated source and the integrity of the time has been maintained. The client and server will share a key specified in /etc/chrony.keys. This method uses a MAC in the NTP packets. The TOE supports SHA1, SHA 512 and SHA256 as the MACs.

FCS_NTP_EXT.1

7.2.4 Random Bit Generation

Random bit generation is provided by the crypto module, RHEL 8.2 OpenSSL v1.1.1c-15-B1, which utilizes CTR_DRBG as defined by NIST SP 800-90A. This is not configurable, and there are no other cryptographic engines provided in the TOE. It has been determined that the entropy source (RDSEED) is sufficiently entropic. OpenSSL uses RDSEED as a source of high quality seeds for generating random numbers. For seeding/reseeding, the seed length used by OpenSSL is at least as much as the output block size, incremented in blocks of 160 bits, in order to ensure sufficient input entropy. Thus, for the AES256 CTR DRBG used by OpenSSL for generating random numbers, the seed length is 320 bits. For key generation in OpenSSL, at least as many random bits as the key length are used, thus ensuring that the keys generated are sufficiently entropic. The minimum entropy assumed is 100%.

FCS_RBG_EXT.1

7.2.5 SSH Client Protocol (Selection-based)

The TOE implements OpenSSH 8.0p1-4, which is conformant to RFC 4251, 4252, 4523, 4554, and 6668. This is not configurable, and there are no other SSH modules or applications provided in the TOE.
The TOE implements and supports SSH-RSA as its only supported public key authentication algorithm. If the TOE is not configured for public key authentication for a specified user, password-based authentication is provided. This is enforced in the OpenSSH configuration files, which are configured at the factory.
The TOE detects packets of size greater than 262144 (256*1024) bytes in the SSH traffic stream and aborts the session when such a packet is discovered. Packet size is detected by reading the size of the payload from the IP packet header. This behaviour is enforced in the OpenSSH and cannot be changed.
The TOE implements SSHv2 transport connections with AES128-CBC and AES256-CBC modes. SSH supports SSH-RSA public key algorithms and no others, rejecting all other public key algorithms. Supported integrity algorithms are HMAC-SHA2-256 and HMAC-SHA2-512, and key exchange is performed using DH-Group14-SHA1 only. This is enforced in the OpenSSH configuration files, which are configured at the factory.
The TOEs SSH configuration also supports a ReKeyLimit parameter. By default, the ReKeyLimit parameter is set by at the factory, and enforces rekeying after 1 hour or 1 GB of data exchanged with the same key. Rekeying is performed upon reaching the threshold that is hit first.
As a server, the ReKeyLimit parameter is enforced in the OpenSSH configuration files, which are configured at the factory. As a client, the ReKeyLimit parameter is enforced in the SSH configuration file in the underlying operating system, which governs all use of the SSH client.
When acting as a client, the TOE authenticates the identity of the SSH server using a local database that associates each server hostname with its corresponding public key file as described in RFC4251 Section 4.1. This is accomplished by using the “known-hosts” file created

by sshd and stored in the underlying filesystem. Should an offered public key not match the information stored in the known-hosts database, the SSH connection will be denied and an audit trail entry will be created to prompt the administrator to verify the information in the known-hosts file, the IP address and hostname of the desired SSH server, and perform any necessary updates to the configuration.
This behaviour is enforced in the OpenSSH configuration files, which are configured at the factory. An administrator may make authorized changes to the known-hosts file.

FCS_SSHC_EXT.1

7.2.6 SSH Server Protocol (Selection-based)

The TOE implements and supports SSH-RSA, RSA-SHA2-256 and RSA-SHA2-512 as its only supported public key authentication algorithms. If the TOE is not configured for public key authentication for a specified user, password-based authentication is provided. This is enforced in the OpenSSH configuration files, which are configured at the factory.

In order to establish a user identity when the SSH client presents a public key for authentication, the client's public key needs to be present in the "authorized_keys" file in the client's home directory on the server, in the ".ssh" sub-directory. When the client presents its public key, the server will compare the key with the copy of the client's key on the server and accept the client as valid only if the keys match.

The TOE detects packets of size greater than 262144 (256*1024) bytes in the SSH traffic stream and aborts the session when such a packet is discovered. Packet size is detected by reading the size of the payload from the IP packet header. This behavior is enforced in the OpenSSH and cannot be changed.

The TOE implements SSHv2 transport connections with AES128-CBC and AES256-CBC modes. SSH supports SSH-RSA, RSA-SHA2-256 and RSA-SHA2-512 public key algorithms and no others, rejecting all other public key algorithms. Supported integrity algorithms are HMAC-SHA2-256 and HMAC-SHA2-512, and key exchange is performed using DH-Group14-SHA1 only. This is enforced in the OpenSSH configuration files, which are configured at the factory.

The TOEs SSH configuration also supports a ReKeyLimit parameter. By default, the ReKeyLimit parameter is set by the factory, and enforces rekeying after 1 hour or 1 GB of data exchanged with the same key. As a server, the ReKeyLimit parameter is enforced in the OpenSSH configuration files, which are configured at the factory. As a client, the ReKeyLimit parameter is enforced in the SSH configuration file in the underlying operating system, which governs all use of the SSH client. Rekeying is performed upon reaching the threshold that is hit first.

FCS_SSHS_EXT.1

7.2.7 TLS Server Protocol Without Mutual Authentication (Selection-based)

The TOE implements TLSv1.2 according to RFC 5246, and supports the following ciphersuites:

- TLS_RSA_with_AES_128_CBC_SHA, defined in RFC 3268
- TLS_ECDHE_RSA_with_AES_256_GCM_SHA384, defined in RFC 5289

The TLS server provides a trusted path for performing administrative functions, during the course of which TLSv1.2 mechanisms are exercised.
This is enforced in the OpenSSL configuration files, which are configured at the factory.

FCS_TLSS_EXT.1

The TOE denies all connection requests from TLS version 1.1 or older, and SSLv3.0 and older. Only TLSv1.2 connections are supported. When a client requests an unsupported version of TLS, the TOE rejects the connection attempt by sending “handshake failure” and “invalid protocol version” error responses to the client. This behavior is enforced in the OpenSSL configuration files, which are configured at the factory.

FCS_TLSS_EXT.1.2

For RSA key agreement in the first supported cipher, the TOE will negotiate key establishment using RSA with a key size of 2048 bits. For ECDHE key agreement in the second supported cipher, the TOE will negotiate key establishment using NIST curve secp256r1. This is enforced in the OpenSSL configuration files, which are configured at the factory.

FCS_TLSS_EXT.1.3

The TOE does not support session resumption or session tickets.

FCS_TLSS_EXT.1.4

7.3 Identification and Authentication

7.3.1 Authentication Failure Management

For each remote administrator, each successive unsuccessful authentication attempt shall be audited and logged. The remote administrator is prevented from successfully logging on to the TOE by the system after a configured number of unsuccessful authentication attempts is reached. Once the number of unsuccessful attempts to authenticate reaches the configured value, the user will be locked out for a certain time interval (10 minutes by default). The ability to login will be restored by the system after the configured lockout time has elapsed, by default. The administrator account “admin” is configured to not be subjected to account locking when logging in locally via the serial console. This is done to ensure that authentication failures by a remote administrator cannot lead to a situation where no administrator is available, either permanently or temporarily. The locally logged in administrator may unlock his own account before the configured lockout time has elapsed.
If public key authentication is proposed, the TOE will initiate authentication based on ssh-rsa, rsa-sha2-256 and rsa-sha2-512. If authentication is based on public keys, there is no password involved. The client’s public key is presented by the remote software. The session will be rejected immediately in case of failure. Therefore, there is no lockout protocol as for password-based authentication.

FIA_AFL.1

7.3.2 Password Management

Passwords created for TOE authentication may be composed of all printable ASCII characters in UTF-8 formatting. For local console and SSH sessions, the administrative password length is configurable by changing the underlying RHEL 8.2 password policy and may be configured to be between 9 and 128 characters. Such password policies are managed by the authenticated administrator and are enforced by the login module.

Modifying the TOE password policy is done by changing the underlying RHEL 8.2 password policy. Authentication to the TOE is done by specifying a valid username and its associated password, which are identical between local and remote authentication.

FIA_PMG_EXT.1

7.3.3 User Identification and Authentication

The only supported authentication mechanisms are via the trusted path provided by SSH, TLS protected interface, or the local console.
For SSH remote administrative sessions, an administrator must connect to the TOE using their SSH client, which will negotiate a secure connection using the supported cryptographic operations.
For TLS remote administrative sessions, an administrator must connect to the TOE using a TLS client, which will negotiate a secure connection using the supported cryptographic operation.
During SSH or TLS remote administrative sessions, the administrator must authenticate using their username and password, or public key (for SSH sessions).
A successful login will be denoted by obtaining a command prompt; unsuccessful logons will result in the connection being dropped by the TOE. Administrators authenticate by providing their username and password, or by use of SSH-RSA public keys if configured. For password authentication, the provided password is hashed according to the underlying RHEL authentication mechanism, and the resultant hash is compared against the stored value for the user's hashed password. If the hashes are the same, authentication for that user is successful. If the hash values are different, authentication fails.
If public key authentication is proposed, the TOE will initiate authentication based on ssh-rsa, rsa-sha2-256 and rsa-sha2-512. If authentication is based on public keys, there is no password involved. The client's public key is presented by the remote software. The session will be rejected immediately in case of failure. Therefore, there is no lockout protocol as for password-based authentication.

For Local administrative sessions, administrators authenticate by providing their username and password at the logon prompt. Successful authentication is denoted by obtaining a command prompt, while unsuccessful authentication results in a prompt to reauthenticate.
No administrative actions or functions are available prior to log in. Only those pre-authentication functions described below are permitted before forcing the user or non-TOE entity to authenticate:

- Display the warning banner in accordance with FTA_TAB.1;
- Respond to ICMP Echo Request, respond to ARP requests with ARP replies, establish TLS connection on TCP port 27777, automated generation of cryptographic keys

FIA_UIA_EXT.1
FIA_UIA_EXT.1.2

7.3.4 Password-based Authentication Mechanism

The TOE provides a local password-based authentication mechanism to identify and verify users before allowing them to perform actions or execute commands on the TOE for both local and remote administrative sessions. This is provided by the underlying RHEL 8.2 user authentication component, which stores authentication data for remote and local console sessions in non-plaintext form in the /etc/shadow file in the underlying filesystem.

FIA_UAU_EXT.2

7.3.5 Protected Authentication Feedback

During authentication, the TOE obscures passwords by failing to echo any information back to the screen. This protects the administrator authentication data by revealing neither the content nor any related data regarding the administrator credentials (such as length or complexity).

FIA_UAU.7

7.3.6 X.509 Certificate Validation

The instances when Certificate validation occurs are listed below:

- when any certificate is imported or installed
- when the server is restarted after being stopped for any reason.

The TOE validates x509v3 certificates according to the validation rules in RFC 5280, terminating with a trusted CA certificate. Certificates presented for validation are validated via CRL as specified in RFC 5280. Certificate Validation of server certificates do not happen when a client connects. Certificate revocation check takes place as a part of Certificate validation.

Such certificate validations are carried out using the OpenSSL module, and this behavior is not configurable by the administrator. While only PEM encoding is supported for certificates, both PEM and DER encodings are supported for CRLs.

FIA_X509_EXT.1.1/Rev (Selection-based)
FIA_X509_EXT.1.2/Rev (Selection-based)

7.3.7 X.509 Certificate Authentication (Selection-based)

The TOE allows administrators to install x509v3 certificates for use in negotiating TLS connections, and [T1] specifies how such certificates are installed. Only one certificate may be installed at any time.

When a connection cannot be established during the validity check of a certificate used in establishing a trusted channel, the certificate will be considered invalid. The only time a connection is established while validating certificates is when a CRL is downloaded. If the connection to download a CRL cannot be established, the certificate will be considered invalid.

FIA_X509_EXT.2.1
FIA_X509_EXT.2.2

7.3.8 X.509 Certificate Requests (Selection-based)

An administrator may generate a Certificate Request Message as specified by RFC 2986, providing the following information in the request: public key, Common Name, Organization, Organizational Unit, and Country.

FIA_X509_EXT.3.1
FIA_X509_EXT.3.2

7.4 Security Management

7.4.1 Management of Security Functions Behaviour

The TOE does not allow any but authorized and authenticated administrators to perform a manual update of the TOE firmware. Unless logged in to a local or remote administrative

session, the commands required to initiate such updates are not available. In order to initiate the update process, the administrator must copy the candidate image to a specific file location within the TOE. Once the copy is completed, the administrator may issue the command to initiate the update process. A local administrative session may be set up via the local serial port, by the Authorized Administrator, with the Bivio provided console cable using the parameters described in the Guidance.

In order to modify the behavior of transmitting audit data to an external IT entity, the security administrator configures the ssh-tunnel-client to establish an encrypted tunnel from the TOE to the external IT entity. Modifying the behavior of the TSF involves changing the config file. An audit log will be generated depicting the modification in the file. User can view the modified files to determine the behavior. The local audit daemon is configured to send audit logs to the local syslog in addition to its own local store, and the syslog daemon is configured to transmit the audit logs via the encrypted tunnel to the remote IT entity. Once configured, the encrypted tunnel is automatically established whenever the TOE boots up. If the remote entity is not available, the TOE will keep retrying the connection until the remote entity is available.

The TOE does not allow any but authorized and authenticated administrators to view or modify any security related settings, or in fact any settings at all. There are no options to manage the TOE or modify its behavior in any way prior to authenticating as an administrator.
After authentication, the administrator may modify the behaviour of the TSF, such as shutting down or rebooting the TOE, replacing the persistent cryptographic keys in use by the TOE, and/or changing the allowed or enabled ciphers for SSH and/or TLS.
The TOE does not allow any but authorized and authenticated administrators to view or modify any services or functions. There are no options to manage the TOE or configure, start, or stop services in any way prior to authenticating as an administrator.
After authentication, the Security administrator is able to start and stop any services offered by the TSF, such as the TLS service, the SSH service, the NTP service, the syslog service, and any background processes running in the underlying RHEL operating system. The administrator may also shutdown or reboot the TOE. These are performed via the Linux systemd service mechanism, except for the TLS login service and the SSH tunnel service. The TLS login service and SSH tunnel service is started and stopped using a special command provided by Bivio and is different from the Linux commands used for other services. The actual daemons are started by system processes running as root.

FMT_MOF.1/Functions, FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services

7.4.2 Management of TSF Data

The TOE does not allow any but authorized and authenticated administrators to view or modify the TSF data. There is no mechanism to interact with any TSF data at all prior to authenticating as an administrator.

FMT_MTD.1/CoreData

The TOE does allow authorized and authenticated administrators to create, import, and delete cryptographic keys. All security-related functions are performed by the authorized administrator(s). All actions related to cryptographic keys require first authenticating as an authorized administrator, and there are no mechanisms to interact with these keys prior to authenticating.

The TOE supports cryptographic keys used by sshd, the ssh tunnel client, and the TLS server. The options that the administrator can perform for management of cryptographic keys for sshd and the ssh tunnel client are generate the keys, replace existing keys, and destroy the keys. In addition, for the TLS server, certificates may be imported into the TOE, and validated. Zeroization of keys can be performed by executing the “zeroize-keyfile” command by the administrator when such keys are no longer required.

FMT_MTD.1/CryptoKeys (Optional)

7.4.3 Specification of Management Functions

When authenticated as an authorized administrator via a local session, the administrator has the ability to perform the following functions:

- Configure the access banner
- Configure the session inactivity time before session termination
- Update the TOE, and to verify the updates using hash comparison capability prior to installing those updates
- Configure the cryptographic functionality
- Start, stop, and restart services
- Ability to modify the behaviour of the transmission of audit data to an external IT entity
- Re-enable an administrator account
- Configure NTP
- Ability to set the time which is used for time-stamps
- Ability to manage the cryptographic keys
- Ability to manage the TOE’s trust store and designate X509.v3 certificates as trust anchors
- Ability to import X.509v3 certificates to the TOE’s trust store
- Ability to configure thresholds for SSH rekeying
- Ability to configure the authentication failure parameters for FIA_AFL.1

When authenticated as an authorized administrator via a remote session, the administrator has the ability to perform the following functions:

- Configure the access banner
- Configure the session inactivity time before session termination or locking
- Update the TOE, and to verify the updates using hash comparison capability prior to installing those updates
- Configure the cryptographic functionality
- Start, stop, and restart services
- Ability to modify the behaviour of the transmission of audit data to an external IT entity
- Configure NTP
- Ability to set the time which is used for time-stamps
- Ability to manage the cryptographic keys
- Ability to manage the TOE’s trust store and designate X509.v3 certificates as trust anchors
- Ability to import X.509v3 certificates to the TOE’s trust store
- Ability to configure thresholds for SSH rekeying
- Ability to configure the authentication failure parameters for FIA_AFL.1

FMT_SMF.1

7.4.4 Restrictions on Security Roles

The TOE maintains the role of “administrator”. There are no other roles. Only one security administrator is required on the system. There are no other users. The administrator is also a security administrator for the purposes of this PP.
Since there is no cause to create a non-administrative user of the TOE, there are no other roles available. The user of the TOE’s administrative and management functions is an authorized administrator, and only a person designated as an authorized administrator may have a user account on the TOE.
The administrator may administer the TOE locally and remotely.
FMT_SMR.2

7.5 Protection of the TSF

7.5.1 Protection of Administrator Passwords

The administrator password is subject to the requirements of FPT_APW_EXT.1. All passwords are stored in SHA512 hashed form, and there is no mechanism provided to read or display administrative password or authentication material.

FPT_APW_EXT.1

7.5.2 TSF Testing

The TOE performs a suite of self-tests upon start up or power on, and periodically during normal operation, and at the request of an authorized administrator. The TOE verifies that:

- The memory/RAM is functioning by writing known values to each register and reading these values back.
- RDSEED is responding as expected by performing the entropy health checking as prescribed in NIST SP 800-90B section 4.
- Software Integrity is valid, by computing the hash of the static system binaries and configuration files and comparing them to a stored value. If the values are identical, then none of the system binaries have changed and software integrity is intact.
- The cryptographic module is performing as expected, by executing Known Answer Tests for each algorithm. These tests are accomplished by encrypting a known value with each cipher, then decrypting it to verify that the decrypted value is as expected.

All tests are performed at startup, and the cryptographic tests may be requested during runtime by the TOE itself or at the request of an authorized administrator. The TSF may initiate periodic runtime cryptographic testing before performing any cryptographic operation (as required by the OpenSSL module when operating in FIPS mode).
Software integrity is checked periodically during runtime by the AIDE package (Advanced Intrusion Detection Environment, which is a file and directory integrity checker provided by the underlying RHEL OS). AIDE is initialized by computing the cryptographic hashes of the TOE configuration and executable files and storing these in a local database. During runtime, the AIDE package performs periodic cryptographic hash computation and comparison of these files against the initialized values stored in the local database.
In total, these tests ensure that the TSF is operating correctly at all times (having demonstrated that memory is operating as expected, the cryptographic module is operating correctly, none of the executable or configuration files have been modified, and that the entropy source is operating correctly). There are no other security-relevant TOE components to test. Should any of these tests fail, the administrator will be notified via alerts (syslog messages).

FPT_TST_EXT.1

7.5.3 Trusted Update

Any software installed on the TOE will become active immediately, or after a reboot if that is required by the update process. If the reboot is requested, the user does not have an option to continue without rebooting. Thus, there is no delayed activation mechanism on the TOE.

The TOE allows the administrator to initiate the update process by downloading the update file from Bivio and following the specific instructions in the administrative guidance. No other mechanism is provided to update the TOE. The TOE also allows the administrator to query the currently running version of software at start up time.

The updates can be only for parts of the system software. The command “rpm -qa” prints out the package names and versions of each individual component (rpm) which is part of the whole installation. The version and build number of BiviOS is within the file /etc/NRDIST and it can be seen by printing out this file.

The administrator is instructed in [T1] to verify the published hash of the downloaded update file before installation. Verification of the published hash is done by executing a command to cause the TOE to generate the hash of the update candidate image stored in the underlying filesystem. This hash is displayed to the administrator for comparison against the published hash available on the Bivio website.

In order to download the published hash value, hash compare and install the update, an active authorization by the Security Administrator is necessary.
When the administrator verifies that the hashes are correct, the update process proceeds. If the administrator indicates that the hashes are not correct, the administrator must contact Bivio support to resolve the error.
When querying the running version of the firmware, the TOE reports the major version and the identification of the latest patch which was installed. Patches have an incrementing sequence number and must be installed in that sequence.

FPT_TUD_EXT.1

7.5.4 Protection of TSF Data

Symmetric keys, private keys, and other CSPs are stored as protected files using the security permissions of the underlying file structure. There are no mechanisms that would allow an administrator to view these keys directly.
Administrative login passwords are stored as hashes that are not readable by anyone. During the login process, the plaintext password entered by the user is hashed, and the resulting hash compared to the stored hash in the password storage file in the underlying file structure. A successful match grants login, while and unsuccessful match results in authentication failure.

FPT_SKP_EXT.1

7.5.5 Reliable Time Stamps

The TOE provides reliable time stamps for all operations. TSF functions that rely on accurate time are NTP, SSH and TLS negotiations (for verifying validity dates of keys or certificates, or the generating of time-based nonce), audit log timestamps, session timeouts, and X.509v3

Certificate verification (for verifying validity of certificates, i.e., the certificate has not expired). All time stamps are provided by the system clock in the underlying RHEL 8.2 operating system.

The TOE can receive timestamps from an external NTP server for clock synchronization. The administrator can set up date and time manually. If NTP is used for clock synchronization, no manual clock adjustments will be necessary. If NTP is not used (not recommended), the administrator is instructed to configure the time at least once per year.

Time is maintained and considered reliable by the hardware clock, which has been measured to provide less than 1.5 seconds of drift per calendar year.

FPT_STM_EXT.1
FPT_STM_EXT.1.2

7.6 TOE Access

7.6.1 TSF-initiated Session Locking

Local administrative sessions are terminated after an administrator-specified period of inactivity. This is configured using the TMOUT global environment variable and is configurable by an administrator. To enforce the timeout, a count-up timer is started when an administrator logs in. After every action taken in an administrative session, the timer is reset. When the timer reaches the stored timeout value in seconds, the session is terminated.

FTA_SSL_EXT.1

7.6.2 TSF-initiated Termination

Remote administrative sessions, both SSH and TLS, are terminated after an administrator-specified period of inactivity. This is configured using the TMOUT global environment variable and is configurable by an administrator. To enforce the timeout, a count-up timer is started when an administrator logs in. After every action taken in an administrative session, the timer is reset. When the timer reaches the stored timeout value in seconds, the session is terminated.

FTA_SSL.3

7.6.3 User-initiated Termination

Users may end their own sessions at any time by use of the “logout” command, by closing their SSH client, or by ending their SSH session.

Remote administrative session termination for TLS sessions works in the same way as for SSH sessions. The administrator may end their own sessions at any time by the use of the “logout” command, or by closing or ending their TLS client in some way.

FTA_SSL.4

7.6.4 Default TOE Access Banners

The available administrative methods of access for remote console session are TLS and SSH administration. The TOE allows the listed remote console sessions and local console session for administrative access. All administrative methods of access (local and remote) are available to the Security Administrator.
Whenever a user attempts to initiate an administrative session, they will be shown the administrator-configurable TOE Access and Warning Banner.

FTA_TAB.1

7.7 Trusted Path/Channels

7.7.1 Inter-TSF Trusted Channel

The TOE supports and enforces trusted channels that protect the communications between the TOE and a remote audit server from unauthorized disclosure or modification. It also supports trusted paths between itself and remote administrators so that the contents of administrative sessions are protected against unauthorized disclosure or modification.

For the Trusted Path (used for administrator login), the TOE will act as a server (SSH server for an SSH based login, and TLS server for a TLS based login). For the Trusted Channel (used to connect to a remote audit server) the TOE will act as an SSH client.

The TOE achieves trusted channels by use of the SSHv2 Protocol which ensures the confidentiality and integrity of communication with the remote audit server. The TOE will initiate the connection, and mutual identification of the endpoints is guaranteed by using public key or password-based authentication for SSH. The SSHv2 protocol ensures that the data transmitted over an SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol. Full details regarding the allowed protocol options are discussed in Sections 7.2.5 and 7.2.6 of this document. In implementing the trusted path, the TOE acts as an SSH client.

The TOE implements trusted paths by use of the SSHv2 protocol and/or TLSv1.2, which ensures the confidentiality and integrity of administrative sessions. The encrypted communication path between the TSF and a remote administrator is provided by the use of an SSH or TLSv1.2 session. Remote administrators initiate the connection to the TOE for both SSH and TLS connections. Assured identification of the endpoints is provided by using public key or password-based authentication for SSH. For TLSv1.2, the administrator authenticates the TOE via X.509v3 certificates, while the TOE authenticates the administrator with a password. The SSHv2 protocol ensures that data transmitted over an SSH session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol. The TLSv1.2 protocol ensures that data transmitted over the TLS session cannot be disclosed or altered by using the encryption and integrity mechanisms of the supported ciphersuites as part of the TLSv1.2 protocol.
Local console access is gained using the Bivio-provided console cable between the console port on the TOE and the administrator workstation with an RS-232 port.

FTP_ITC.1 & FTP_TRP.1/Admin

8. Terms and Definitions

Table 7: TOE Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
BIOS	Basic Input/Output System
CA	Certificate Authority
CBC	Cipher Block Chaining
CLI	Command Line Interface
CMOS	Complementary Metal–Oxide–Semiconductor
CRL	Certificate Revocation List
CSP	Critical Security Parameter
CSR	Certificate Signing Request
CTR	Counter
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
DTLS	Datagram Transport Layer Security
ECDH	Elliptic Curve Diffie-Hellman
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
ESP	Encapsulating Security Payload
GCM	Galois Counter Mode
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPs	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
KAT	Known Answer Test
KDF	Key Derivation Function
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PCT	Pairwise Consistency Test
PKCS	Public Key Cryptography Standards
RFC	Requests for Comments
RSA	Rivest-Shamir-Adleman
SA	Security Association
SAN	Subject Alternative Name
SFP	Small Form-Factor Pluggable
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SPD	Security Policy Database
SSH	Secure Shell
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UI	User Interface

Table 7: TOE Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
URI	Uniform Resource Identifier
VPN	Virtual Private Network

Table 8: CC Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
CC	Common Criteria
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
DOD	Department of Defense
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standards
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Function Policy
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification

9. References

Table 9: TOE Guidance Documentation			
Reference	Description	Version	Date
[T1]	Bivio 6310-NC Common Criteria Administrative Guidance, Version 1.10	V1.10	Nov 23, 2020

Table 10: Common Criteria v3.1 References			
Reference	Description	Version	Date
[C1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2017-04-001	V3.1 R5	April 2017
[C2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components CCMB-2017-04-002	V3.1 R5	April 2017
[C3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components CCMB-2017-04-003	V3.1 R5	April 2017
[C4]	Common Criteria for Information Technology Security Evaluation Evaluation Methodology CCMB-2017-04-004	V3.1 R5	April 2017
[C5]	CC and CEM addenda - Exact Conformance, Selection-Based SFRs, Optional SFRs CCDB-2017-05-xxx	V0.5	May 2017

Table 11: Supporting Documentation			
Reference	Description	Version	Date
[cPP]	Collaborative Protection Profile for Network Devices	2.2e	March 23, 2020
[SD]	Supporting Document Mandatory Technical Document Evaluation Activities for Network Device cPP	2.2	December 2019