KlasOS Keel 5.4.0 Security Target

Document Version: 1.5, July 5th 2024

intertek acumen security

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History

Version	Date	Changes
0.1	April 22, 2022	Initial Release.
0.2	June 29, 2022	Removed SSH client, updated NIAP TDs.
0.3	July 13, 2022	Update to CAVP algorithm details.
0.4	September 21, 2022	TOE name corrected to VoyagerVM or VoyagerVMm (space removed. Updated technical decision table (added TD0631, TD0638, TD0639, TD0670) and also added column for PP. Added FCS_TLSC_EXT.1 to table of auditable events. Incorporated comments from QA review. Iterated FTP_ITC.1 to cover audit server and SD-WAN.
0.5	February 22, 2023	Change to description of SD-WAN connection to include TSF data transfers (1.3.1 and TSS Table 16). Minor typos corrected in FCS_DTLSC_EXT.1.1 and FMT_MOF.1.1
0.6	July 24, 2023	Updated ST Claims
0.7	August 10, 2023	Updated CAVP algorithm details
0.8	November 13, 2023	Updated Claims and addressed ECR Comments
0.9	December 20, 2023	Updated Claims and addressed ECR Comments
1.0	February 21, 2024	Updated TSS and claims
1.1	February 26, 2024	Implemented Vendor feedback and updated TD table
1.2	March 4, 2024	Updated TSS
1.3	April 23, 2024	Updated Document References and Minor Updates
1.4	June 3, 2024	Resolved QA and Technical review feedback
1.5	July 5th, 2024	Updated to address ECR comments

Contents

1 Introduction

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

1.1 Security Target and TOE Reference

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

Table 1 – TOE/ST Identification

Category	Identifier
ST Title	KlasOS Keel 5.4.0 Security Target
ST Version	1.5
ST Date	July 5th, 2024
ST Author	Acumen Security, LLC.
TOE Identifier	KlasOS Keel
TOE Version	5.4.0
TOE Developer	Klas
Key Words	Network Device, Firewall, DTLS, SDWAN

1.2 TOE Overview

The TOE is KlasOS Keel 5.4.0 running on the VoyagerVMm, TRX R2 and Voyager VM3.0 platforms (herein referred to as the TOE). It runs the KlasOS Keel 5.4.0 firmware combining both connectivity and local compute capabilities. Network connectivity includes ethernet and SDWAN. Computing and firewall capabilities are combined in one unit. This provides users with cloud connectivity when necessary and local processing power for analytics when there is no backhaul. Administration can be performed locally or over a trusted SSH channel.

1.3 TOE Description

This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references. All TOE models below run the same Klas Keel 5.4.0 binary file.

Table 2 – TOE Models

TOE Model	Specifications
VoyagerVMm (i3) and VoyagerVMm (i5) 	5th Gen Intel® Dual Core i3-5010U (1.8 GHz) Broadwell-U, 8 GB DDR3 RAM Network Ports: 1 x console, 4 x Gb Ethernet Storage: Samsung 850 EVO 256 GB mSATA SSD or Samsung 1TB mSATA SSD
	5th Gen Intel® Quad Core i5-5350U (1.8 GHz) Broadwell-U, 32 GB DDR3 RAM Network Ports: 1 x console, 4 x Gb Ethernet

TOE Model	Specifications
	Storage: Samsung 850 EVO 256 GB mSATA SSD or Samsung 1TB mSATA SSD
TRX R2 (4-core) and TRX R2 (8-core) 	Atom™/Denverton C3508 Intel® Atom™ Denverton C3508 4-Core processor with 1.6 GHz clock. 8 GB RAM (upgradeable to 32 GB) Network Ports: 2 x 1 Gb Ethernet 4G/LTE Modems Sierra Wireless EM7455 LTE Cat-6 (B1, B2, B3, B4, B5, B7, B12, B13, B20, B25, B26, B29, B30, B41) Sierra Wireless EM7511 LTE Cat-12 (B1, B2, B3, B4, B5, B7, B8, B9, B12, B13, B14, B18, B19, B20, B26, B29, B30, B32, B41, B42, B43, B46, B48, B66) IEEE802.11 ac/b/g/n 3×3 MIMO Wi-Fi modem with data rates of 1.3 Gb/s downlink in 2.4/5 Ghz bands
	Atom™/Denverton C3708 Intel® Atom™ Denverton C3708 8-Core processor with 1.7 GHz clock. 8 GB RAM (upgradeable to 32 GB) Network Ports: 2 x 1 Gb Ethernet 4G/LTE Modems Sierra Wireless EM7455 LTE Cat-6 (B1, B2, B3, B4, B5, B7, B12, B13, B20, B25, B26, B29, B30, B41) Sierra Wireless EM7511 LTE Cat-12 (B1, B2, B3, B4, B5, B7, B8, B9, B12, B13, B14, B18, B19, B20, B26, B29, B30, B32, B41, B42, B43, B46, B48, B66) IEEE802.11 ac/b/g/n 3×3 MIMO Wi-Fi modem with data rates of 1.3 Gb/s downlink in 2.4/5 Ghz bands
VoyagerVM 3.0 	Xeon D-1539 Intel® Xeon Processor D1539 16-Core with 48 or 96 GB RAM Network Ports: 1 x console, 2 x 10 GB SFP, 2 x 1GB ethernet Storage: removable SATA dual SSDs, removable NVMe Voyager Ignition Key (VIK+)
	Xeon D-1559 Intel® Xeon Processor D1559 12-Core with 48 or 96 GB RAM Network Ports: 1 x console, 2 x 10 GB SFP, 2 x 1GB ethernet Storage: removable SATA dual SSDs, removable NVMe Voyager Ignition Key (VIK+)
	Xeon D-1577 Intel® Xeon Processor D1577 16-Core with 48 or 96 GB RAM

TOE Model	Specifications
	Network Ports: 1 x console, 2 x 10 GB SFP, 2 x 1GB ethernet Storage: removable SATA dual SSDs, removable NVMe Voyager Ignition Key (VIK+)

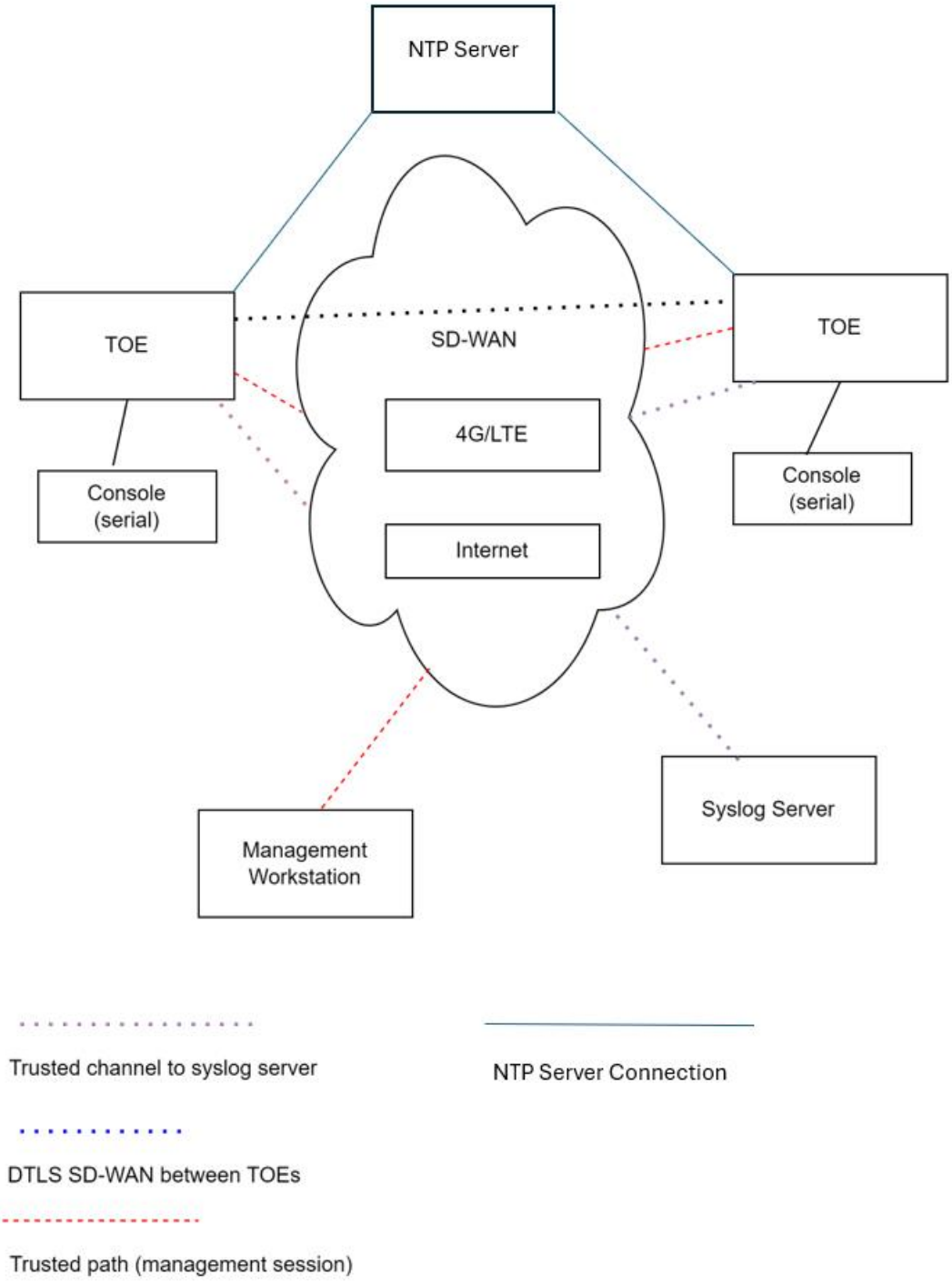

Figure 1 – Representative TOE Deployment

1.3.1 Physical Boundaries

The TOE boundary is the hardware appliance which is comprised of hardware and the KlasOS Keel software component. The TOE hardware models are provided in Table 2 – TOE Models.

The TOE also supports connection to one or more TOEs over an SD-WAN, which is protected by DTLS. In the evaluated configuration, this connection is used solely for the administration of another TOE using SSH over the SD-WAN connection.

The TOE also supports secure connectivity with several other IT environment devices, including the ones identified in the following table.

The TOE implements HTTPS as a limited functionality GUI back to the management workstation. The GUI only offers basic monitoring capabilities and is secured via TLS when an administrator is logged in. Peer certificates are not required for authentication.

Table 3 – IT Environment Components

Component	Required	Purpose/Description
Local Management Workstation	Yes	A management workstation that is directly connected to the TOE's console port may be used by the TOE administrator to support TOE administration.
Remote Management Workstation / SSH Client	Yes	This includes any IT Environment Management workstation with a SSH client installed that is used by the TOE administrator to support TOE administration through SSH protected channel. Any SSH client that supports SSHv2 may be used. This remote management station is also utilized to access the TOE's HTTPS GUI for monitoring capabilities.
Syslog Server	Yes	The syslog audit server is used for remote storage of audit records that have been generated by and transmitted from the TOE. An SSH tunnel is established by the TOE and logs are transmitted using this encrypted method.
NTP Server	No	The NTP server is used to send reliable timestamps to the TOE using NTPv3 and SHA1 as the message digest algorithm.

1.3.2 Security Functions Provided by the TOE

The TOE provides the security functions required by the Collaborative Protection Profile for Network Devices, hereafter referred to as NDcPP v2.2e or NDcPP. In addition, the TOE provides security functions for the PP-Configuration for Network Devices and Stateful Traffic Filter Firewalls. The TOE implements the following security requirements:

- Security Audit (FAU)
- Cryptographic Support (FCS)

- User Data Protection (FDP)
- Firewall (FFW)
- Identification and Authentication (FIA)
- Security Management (FMT)
- Protection of the TSF (FPT)
- TOE Access (FTA)
- Trusted Path/Channels (FTP)

1.3.2.1 Security Audit (FAU)

The TOE generates audit events for all start-up and shutdown functions as well as all auditable events specified in Table 13 – Security Functional Requirements and Auditable Events. Audit events are also generated for management actions specified in FAU_GEN.1. The TOE stores audit records locally and can export them to an external syslog server using SSHv2 as a tunnel. Each audit record contains the date and time of the event, type of event, subject identity, and other relevant data for the event. Only a security administrator can enable logging to a syslog server.

1.3.2.2 Cryptographic Support (FCS)

The cryptographic used in the TOE are presented in the following table.

Table 4 –TOE Cryptography Implementation

Cryptographic Methods	Usage
FCS_CKM.1 Cryptographic Key Generation	• 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’ [selection: P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4; • FFC Schemes using ‘safe-prime’ groups that meet the following: “NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [RFC 3526, RFC 7919].
FCS_CKM.2 Cryptographic Key Establishment	• RSA-based key establishment conforming to 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 • Elliptical curve-based establishment conforming to NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”; • FFC Schemes using “safe-prime” groups that meet the following: ‘NIST Special Publication 800-56A Revision 3, “Recommendation for

Cryptographic Methods	Usage
	Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 3526].
FCS_CKM.4 Cryptographic Key Destruction	• Refer to Table 19 – Key Storage and Zeroization for Key Zeroization details.
FCS_COP.1/DataEncryption Cryptographic Operations (AES Data Encryption/Decryption)	• AES encryption and decryption conforming to CBC, CTR and GCM as specified in ISO 10116. • AES key size supported is 128 and 256 bits • AES modes supported are CBC, CTR and GCM.
FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)	• Cryptographic hashing services conforming to ISO/IEC 10118-3:2004. • Hashing algorithms supported are: SHA-1, SHA-256, SHA-384, and SHA-512. • Message digest sizes supported are: 160, 256, 384, and 512 bits.
FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)	• Keyed-hash message authentication conforming to ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm2". • Keyed hash algorithm supported are: HMAC-SHA1, HMAC-SHA-256, HMAC-SHA384, and HMAC-SHA-512 • Key sizes supported are: 160, 256, 384 and 512 bits. • Message digest sizes supported are: 160, 256, 384, and 512 bits.
FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)	• RSA digital signature algorithm conforming to 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. • RSA key sizes supported are: 2048, 3072, and 4096 bits. • Elliptical curve digital signature algorithm conforming to FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing NIST curves ISO/IEC 14888-3, Section 6.4. • Elliptical curve key sizes supported are 256, 384 and 521 bits.
FCS_DTLSC_EXT.1 DTLS Client Protocol without Mutual Authentication	• The TOE supports DTLS version 1.2 for secure communication between TOEs.

Cryptographic Methods	Usage
FCS_DTLSC_EXT.2 DTLS Client Protocol with Mutual Authentication	• The TOE supports DTLS version 1.2 for secure communication between TOEs using mutual authentication.
FCS_DTLSS_EXT.1 DTLS Server Protocol without Mutual Authentication	• The TOE supports DTLS version 1.2 for secure communication between TOEs.
FCS_DTLSS_EXT.2 DTLS Server Protocol with Mutual Authentication	• The TOE supports DTLS version 1.2 for secure communication between TOEs using mutual authentication
FCS_HTTPS_EXT.1 HTTPS Protocol	• The TOE supports HTTPS using TLS and complies with RFC 2818.
FCS_NTP_EXT.1 NTP Protocol	• The TOE supports NTP v3 and adheres to RFC 1305. • Authentication is performed using SHA-1 as the message digest algorithm.
FCS_RBG_EXT.1 Random Bit Generation	• Random number generation conforming to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions” • The TOE leverages CTR_DRBG(AES) • CTR_DRBG seeded with a minimum of 256 bits of entropy.
FCS_SSHS_EXT.1 SSH Server Protocol	• The TOE supports SSH v2 protocol compliant to the following RFCs:4251, 4252, 4253, 4254, 5656, and 6668. • The TOE supports password-based and public-key-based authentication. • SSH public-key authentication uses ssh-rsa, ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384. • SSH transport uses the following encryption algorithms: aes128-cbc, and aes256-cbc. • Packets greater than 33,292 bytes in an SSH transport connection are dropped. • SSH transport uses the following data integrity MAC algorithms: hmac-sha1, hmac-sha2-256, hmac-sha384, and hmac-sha2-512 • Key exchange algorithms supported are: diffie-hellman-group14- sha1, ecdh-sha2-nistp256 and ecdh-sha2-nistp384. • The TOE ensures that during SSH connections, the same session keys are used for a threshold of no longer than one hour and no more than one gigabyte of transmitted data.
FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication	• The TOE supports TLS 1.2 (RFC 5246) for HTTPS connections

1.3.2.3 User Data Protection (FDP)

For firewall traffic flowing through the TOE any previous information is made unavailable when a new resource is required to be allocated. This ensures that data is not inadvertently sent to an unintended recipient.

1.3.2.4 Firewall (FFW)

The rules allow traffic traversing the TOE to be permitted or dropped and the administrator can choose whether logging occurs when the rule's conditions are met.

1.3.2.5 Identification and Authentication (FIA)

All users must be authenticated by the TOE prior to carrying out any administrative actions. The TOE supports password-based and public-key based authentication. An administrator can set a minimum password length on the TOE which must be at least 15 characters.

1.3.2.6 Security Management (FMT)

The TOE supports local and remote management of its security functions including:

- Ability to configure the access banner
- Ability to configure the session inactivity time before session termination or locking
- Ability to update the TOE, and to verify the updates using [*digital signature*] capability prior to installing those updates
- Ability to configure the authentication failure parameters for FIA_AFL.1
- Ability to 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 re-enable an Administrator account
- Ability to set the time which is used for time-stamps
- Ability to configure NTP
- Ability to configure the reference identifier for the peer
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors
- Ability to import X.509v3 certificates to the TOE's trust store
- Ability to manage the trusted public keys database
- Ability to configure firewall rules

The administrative user can perform all the above security-related management functions.

1.3.2.7 Protection of the TSF (FPT)

The TOE protects all passwords, pre-shared keys, symmetric keys, and private keys from unauthorized disclosure. Passwords are stored as SHA 512 hashes. The TOE executes self-tests during initial start-up to ensure correct operation and enforcement of its security functions. The TOE internally maintains the date and time. An administrator can install software updates to the TOE after they are verified using a digital signature mechanism.

1.3.2.8 TOE Access (FTA)

The TOE displays a customizable banner before any administrative session can be established with it. The TOE will terminate local or remote interactive sessions after a specified period of session inactivity configured by an administrator. An administrator can terminate their own interactive local or remote sessions.

1.3.2.9 Trusted Path/Channels (FTP)

The TOE supports SSH for secure communications with authorized IT entities such as syslog servers. The TOE supports SSHv2 (remote CLI) for secure remote administration. The TOE also supports DTLS for secure communication between TOEs to support SD-WAN.

1.3.3 TOE Documentation

The following document is essential to understanding and controlling the TOE in the evaluated configuration:

- KlasOS 5.4.0 Keel Operational User Guidance, Version 0.4, June 2024.

1.4 TOE Environment

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

Table 5 – Required Environmental Components

Components	Description
Local Management Workstation	A management workstation that is directly connected to the TOE's console port shall be used by the TOE administrator to support TOE administration locally.
Remote Management Workstation / SSH Client	A management workstation with a SSHv2 client installed shall be used by the TOE administrator to support TOE administration through a SSHv2 protected channel. Any SSH client that supports SSHv2 may be used. This remote management station is also utilized to access the TOE's HTTPS GUI for monitoring capabilities.
Syslog Server	The syslog audit server is used for remote storage of audit records that have been generated by and transmitted from the TOE.

1.5 Product Functionality not Included in the Scope of the Evaluation

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

- SNMP
- Spanning-Tree
- Port Security
- TACACS+
- RADIUS

The TOE has SNMP functionality disabled by default and it should not be enabled for the Common Criteria evaluated configuration.

2 Conformance Claims

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

2.1 CC Conformance Claims

The TOE is conformant to the following:

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

2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- PP-Configuration for Network Devices and Stateful Traffic Filter Firewalls. This PP-Configuration includes the following:
  - collaborative Protection Profile for Network Devices, Version 2.2e [CPP_ND_V2.2E]
  - PP-Module for Stateful Traffic Filter Firewalls, Version 1.4e [MOD_CPP_FW_V1.4E]

2.3 Conformance Rationale

This ST provides exact conformance to the items listed in the previous section. The security problem definition, security objectives, and security requirements in this ST are all taken from the Protection Profile (PP), PP-Modules, and Functional Package listed in Section 2.2 Protection Profile Conformance, performing only the operations defined there.

2.3.1 Technical Decisions

All NIAP TDs issued to date and applicable to CPP_ND_V2.2E and MOD_CPP_FW_V1.4E have been considered. Table 6 identifies all applicable TDs.

Table 6 – Relevant Technical Decisions

Technical Decision	Applicable PP	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	cpp_nd_v2.2e	Y	
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	cpp_nd_v2.2e	Y	
TD0536: NIT Technical Decision for Update Verification Inconsistency	cpp_nd_v2.2e	Y	
TD0537: NIT Technical Decision for Incorrect	cpp_nd_v2.2e	Y	

Technical Decision	Applicable PP	Applicable (Y/N)	Exclusion Rationale (if applicable)
reference to FCS_TLSC_EXT.2.3			
TD0545: NIT Technical Decision for Conflicting FW rules cannot be configured (extension of Rfl#201837)	mod_cpp_fw_v1.4e	Y	
TD0546: NIT Technical Decision for DTLS - clarification of Application Note 63	cpp_nd_v2.2e	Y	
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	cpp_nd_v2.2e	Y	
TD0551: NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata	mod_cpp_fw_v1.4e	Y	
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	cpp_nd_v2.2e	Y	
TD0556: NIT Technical Decision for RFC 5077 question	cpp_nd_v2.2e	Y	
TD0563: NiT Technical Decision for Clarification of audit date information	cpp_nd_v2.2e	Y	
TD0564: NiT Technical Decision for Vulnerability Analysis Search Criteria	cpp_nd_v2.2e	Y	
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	cpp_nd_v2.2e	Y	
TD0570: NiT Technical Decision for Clarification about FIA_AFL.1	cpp_nd_v2.2e	Y	
TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1	cpp_nd_v2.2e	Y	
TD0572: NiT Technical Decision for Restricting	cpp_nd_v2.2e	Y	

Technical Decision	Applicable PP	Applicable (Y/N)	Exclusion Rationale (if applicable)
FTP_ITC.1 to only IP address identifiers			
TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	cpp_nd_v2.2e	Y	
TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	cpp_nd_v2.2e	Y	
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	cpp_nd_v2.2e	N	This TD is not applicable since the TOE is a hardware appliance.
TD0592: NIT Technical Decision for Local Storage of Audit Records	cpp_nd_v2.2e	Y	
TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server	cpp_nd_v2.2e	Y	
TD0632: NIT Technical Decision for Consistency with Time Data for vNDs	cpp_nd_v2.2e	N	This TD is not applicable since the TOE is a hardware appliance.
TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters	cpp_nd_v2.2e	Y	
TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH	cpp_nd_v2.2e	Y	
TD0638: NIT Technical Decision for Key Pair Generation for Authentication	cpp_nd_v2.2e	Y	
TD0639: NIT Technical Decision for Clarification for NTP MAC Keys	cpp_nd_v2.2e	Y	
TD0670: NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	cpp_nd_v2.2e	Y	

Technical Decision	Applicable PP	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0738: NIT Technical Decision for Link to Allowed-With List	cpp_nd_v2.2e	Y	
TD0790: NIT Technical Decision: Clarification Required for testing IPv6	cpp_nd_v2.2e	Y	
TD0792: NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	cpp_nd_v2.2e	Y	
TD0800: Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	cpp_nd_v2.2e	N	This TD is not applicable since the TOE does not support IPsec
TD0827: Aligning MOD_CPP_FW_v1.4E with CPP_ND_V3.0E	MOD_CPP_FW_v1.4e	N	This TD is not applicable, this evaluation is performed for cpp_nd_v2.2e.

3 Security Problem Definition

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

3.1 Threats

The threats included in Table 7 are drawn directly from the PP and any EPs/Modules/Packages specified in Section 2.2.

Table 7 – Threats

ID	Threat
T.MALICIOUS TRAFFIC	An attacker may attempt to send malformed packets to a machine in hopes of causing the network stack or services listening on UDP/TCP ports of the target machine to crash.
T.NETWORK_ACCESS	With knowledge of the services that are exported by machines on a subnet, an attacker may attempt to exploit those services by mounting attacks against those services.
T.NETWORK_DISCLOSURE	An attacker may attempt to “map” a subnet to determine the machines that reside on the network, and obtaining the IP addresses of machines, as well as the services (ports) those machines are offering. This information could be used to mount attacks to those machines via the services that are exported.
T.NETWORK_MISUSE	An attacker may attempt to use services that are exported by machines in a way that is unintended by a site’s security policies. For example, an attacker might be able to use a service to “anonymize” the attacker’s machine as they mount attacks against others.
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_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.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

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

ID	Threat
	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.

3.2 Assumptions

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

Table 8 – Assumptions

ID	Assumption
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.
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).
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).

ID	Assumption
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and aDHEring to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. (The paragraph that follows is for x509v3 cert-based authentication. If not relevant, remove) 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.

3.3 Organizational Security Policies

The OSPs included in Table 9 are drawn directly from the PP and any relevant EPs/Modules/Packages.

Table 9 – OSPs

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

4 Security Objectives

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

4.1 Security Objectives for the TOE

The security objectives in the following table apply to the TOE.

Table 10 – Security Objectives

ID	Security Objectives
O.RESIDUAL_INFORMATION	The TOE shall implement measures to ensure that any previous information content of network packets sent through the TOE is made unavailable either upon deallocation of the memory area containing the network packet or upon allocation of a memory area for a newly arriving network packet or both
O.STATEFUL_TRAFFIC_FILTERING	The TOE shall perform stateful traffic filtering on network packets that it processes. For this the TOE shall support the definition of stateful traffic filtering rules that allow to permit or drop network packets. The TOE shall support assignment of the stateful traffic filtering rules to each distinct network interface. The TOE shall support the processing of the applicable stateful traffic filtering rules in an administratively defined order. The TOE shall deny the flow of network packets if no matching stateful traffic filtering rule is identified. Depending on the implementation, the TOE might support the stateful traffic filtering of Dynamic Protocols (optional).

4.2 Security Objectives for the Operational Environment

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

Table 11 – Security Objectives for the Operational Environment

ID	Objectives for the Operational Environment
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.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.
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.

ID	Objectives for the Operational Environment
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.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.
OE.TRUSTED_ADMN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. 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.

5 Extended Components Definition

5.1 Extended Security Functional Components

All extended components are sourced directly from [PP].

5.2 Extended Security Functional Requirements Rationale

All extended security functional components are sourced directly from [PP]. Exact conformance required by the PP also mandates inclusion of all applicable extended components defined in the PP.

6 Security Requirements

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

Table 12 – SFRs

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG.1	Protected Audit Trail Storage
FAU_STG_EXT.1	Protected Audit Event Storage
FAU_STG_EXT.3/LocSpace	Action in Case of Possible Audit Data Loss
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_DTLSC_EXT.1	DTLS Client Protocol Without Mutual Authentication
FCS_DTLSC_EXT.2	DTLS Client Protocol With Mutual Authentication
FCS_DTLSS_EXT.1	DTLS Server Protocol Without Mutual Authentication
FCS_DTLSS_EXT.2	DTLS Server Protocol with Mutual Authentication
FCS_HTTPS_EXT.1	HTTPS Protocol
FCS_NTP_EXT.1	NTP Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHC_EXT.1	SSH Client Protocol
FCS_SSHS_EXT.1	SSH Server Protocol
FCS_TLSS_EXT.1	TLS Server Protocol
FDP_RIP.2	Full Residual Information Protection
FFW_RUL_EXT.1	Stateful Traffic Filtering
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1/Functions	Management of Security Functions Behaviour

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

6.1 Conventions

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

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

6.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

6.2.1 Security Audit (FAU)

6.2.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1

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

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

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

d) *Specifically defined auditable events listed in Table 13*.

Application Note: This SFR has been updated as per TD0639.

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 13.*

Application Note: This SFR has been updated as per TD0563.

Table 13 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	Start-up of the audit function	None
	Shutdown of the audit function	None
	Administrative Login	Name of user account shall be logged if individual user accounts are required for administrators
	Administrative Logout	
	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 cryptographic keys	In addition to the action itself, a unique key name or key reference shall be logged
	Changing of cryptographic keys	
	Deleting of cryptographic keys	
	Resetting passwords	Name of related user account shall be logged
FAU_GEN.2	None	None
FAU_STG.1	None	None
FAU_STG_EXT.1	None	None
FAU_STG_EXT.3/LocSpace	Low storage space for audit events	None
FCS_CKM.1	None	None
FCS_CKM.2	None	None

Requirement	Auditable Events	Additional Audit Record Contents
FCS_CKM.4	None	None
FCS_COP.1/DataEncryption	None	None
FCS_COP.1/Hash	None	None
FCS_COP.1/KeyedHash	None	None
FCS_COP.1/SigGen	None	None
FCS_DTLSC_EXT.1	Failure to establish a DTLS session	Reason for failure
FCS_DTLSC_EXT.2	Detected replay attacks	Source of the replay attack.
FCS_DTLSS_EXT.1	Failure to establish a DTLS session	Reason for failure
FCS_DTLSS_EXT.1	Detected replay attacks	Identity (e.g., source of IP address) of the source of the replay attack
FCS_DTLSS_EXT.2	Failure to authenticate the client	Reason for failure
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure
FCS_NTP_EXT.1	• Configuration of a new time server • Removal of configured time server	• Identity of new/removed time server
FCS_RBG_EXT.1	None	None
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_SSHC_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FDP_RIP.2	None	None
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	• Source and destination addresses • Source and destination ports • Transport Layer Protocol • TOE Interface
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_UAU_EXT.2	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UAU.7	None	None
FIA_UIA_EXT.1	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)

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

Requirement	Auditable Events	Additional Audit Record Contents
FTP_ITC.1/Audit	• 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_ITC.1/SD-WAN	• 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

6.2.1.2 FAU_GEN.2 User Identity Association

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

6.2.1.3 FAU_STG.1 Protected Audit Trail Storage

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

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

6.2.1.4 FAU_STG_EXT.1 Protected Audit Event Storage

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

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

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

].

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

6.2.1.5 FAU_STG_EXT.3/LocSpace Action in Case of Possible Audit Data Loss

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*.

6.2.2 Cryptographic Support (FCS)

6.2.2.1 FCS_CKM.1 Cryptographic Key Generation

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

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*
- *ECC schemes using 'NIST curves' [selection: P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;*
- *FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526, RFC 7919].*

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

Application Note: This SFR has been updated as per TD0638.

6.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

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

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

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

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

6.2.2.3 FCS_CKM.4 Cryptographic Key Destruction

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

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

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

that meets the following: *No Standard.*

Application Note: This SFR has been updated as per TD0639.

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

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

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

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

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

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

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

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

- *RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048, 3072 and 4096 bits]*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [assignment: 256, 384 or 521 bits]*

]
that meet the following: [

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

].

6.2.2.8 FCS_DTLSC_EXT.1 DTLS Client Protocol without Mutual Authentication

FCS_DTLSC_EXT.1.1
The TSF shall implement [selection: *DTLS 1.2 (RFC 6347)*] supporting the following ciphersuites: [

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*

and no other ciphersuites]

Application Note: This SFR has been updated as per TD0546.

FCS_DTLSC_EXT.1.2
The TSF shall verify that the presented identifier matches [*The reference identifier per RFC 6125 section 6, IPv4 in CN or SAN, IPv6 in CN or SAN, and no other attribute types*].

FCS_DTLSC_EXT.1.3

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

- *Not implement any administrator override mechanism*

].

FCS_DTLSC_EXT.1.4
The TSF shall [*not present the Supported Elliptic Curves/Supported Groups Extension]* in the Client Hello.

6.2.2.9 FCS_DTLSC_EXT.2 DTLS Client Protocol with Mutual Authentication

FCS_DTLSC_EXT.2.1
The TSF shall support mutual authentication using X.509v3 certificates.

FCS_DTLSC_EXT.2.2
The TSF shall [*silently discard the record*] if a message received contains an invalid MAC.

FCS_DTLSC_EXT.2.3
The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received.
- DTLS records too old to fit in the sliding window.

6.2.2.10 FCS_DTLSS_EXT.1 DTLS Server Protocol without Mutual Authentication

FCS_DTLSS_EXT.1.1
The TSF shall implement [selection: *DTLS 1.2 (RFC 6347)*] supporting the following ciphersuites: [

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,*

- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*

] and no other ciphersuites.

FCS_DTLSS_EXT.1.2
The TSF shall deny connections from clients requesting *none*.

FCS_DTLSS_EXT.1.3

The TSF shall not proceed with a connection handshake attempt if the DTLS Client fails validation.

FCS_DTLSS_EXT.1.4
The product shall perform key establishment for DTLS using [

- *RSA with* size [2048 bits, 3072 bits, 4096 bits, no other sizes],
- *no other key establishment methods*

].

FCS_DTLSS_EXT.1.5
The TSF shall [*silently discard the record*] if a message received contains an invalid MAC.

FCS_DTLSS_EXT.1.6
The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received.
- DTLS records too old to fit in the sliding window.

FCS_DTLSS_EXT.1.7
The TSF shall support [*no session resumption or session tickets*].

6.2.2.11 FCS_DTLSS_EXT.2 DTLS Server Protocol with Mutual Authentication

FCS_DTLSS_EXT.2.1 The TSF shall support mutual authentication of DTLS clients using X.509v3 certificates.

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

- *Not implement any administrator override mechanism*

].

FCS_DTLSS_EXT.2.3 The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the client.

6.2.2.12 FCS_HTTPS_EXT.1 HTTPS Protocol

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

FCS_HTTPS_EXT.1.2
The TSF shall implement the HTTPS protocol using TLS.

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

6.2.2.13 FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1.1
The TSF shall use only the following NTP version(s) [*NTP v3 (RFC 1305)*].

FCS_NTP_EXT.1.2
The TSF shall update its system time using [
- Authentication using [*SHA1*] as the message digest algorithm.

].

Application Note: This SFR has been updated as per TD0639.

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.

Application Note: This SFR has been updated as per TD0528.

6.2.2.14 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.

6.2.2.15 FCS_SSHC_EXT.1.1 SSH Client Protocol

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [*5656, 6668*].

FCS_SSHC_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, and [*no other method*].

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

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than *[33,292]* 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, aes128-ctr, aes256-ctr*].

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [*ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384*] as its public key algorithm(s) and rejects all other public key algorithms

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

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [*hmac-sha1, 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, ecdh-sha2-nistp256*] and [*ecdh-sha2-nistp384*] 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.

6.2.2.16 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1
The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [*5656, 6668*].

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

Application Note: This SFR has been updated as per TD0631.

FCS_SSHS_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than *[33,292]* 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, aes128-ctr, aes256-ctr*].

FCS_SSHS_EXT.1.5
The TSF shall ensure that the SSH public-key based authentication implementation uses [*ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384*] as its public key algorithm(s) and rejects all other public key algorithms.

Application Note: This SFR has been updated as per TD0631.

FCS_SSHS_EXT.1.6
The TSF shall ensure that the SSH transport implementation uses [*hmac-sha1, 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, ecdh-sha2-nistp256*] and [*ecdh-sha2-nistp384*] 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.

6.2.2.17 FCS_TLSS_EXT.1 TLS Sever Protocol Without Mutual Authentication

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

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*

] 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 size [*2048 bits, 3072 bits, 4096 bits, no other sizes*],*
- *no other key establishment methods*

].

Application Note: This SFR has been updated as per TD0635.

FCS_TLSS_EXT.1.4 The TSF shall support [*session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2)*].

6.2.3 User Data Protection (FDP)

6.2.3.1 FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1
The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*allocation of the resource to*] all objects.

6.2.4 Firewall (FFW)

6.2.4.1 FFW_RUL_EXT.1 Stateful Traffic Filtering

FFW_RUL_EXT.1.1
The TSF shall perform stateful traffic filtering on network packets processed by the TOE.

FFW_RUL_EXT.1.2
The TSF shall allow the definition of stateful traffic filtering rules using the following network protocol fields:

- *ICMPv4*
  - *Type*
  - *Code*
- *ICMPv6*
  - *Type*
  - *Code*
- *IPv4*
  - *Source address*
  - *Destination Address*
  - *Transport Layer Protocol*
- *IPv6*
  - *Source address*
  - *Destination Address*
  - *Transport Layer Protocol*
  - *[no other field]*
- *TCP*
  - *Source Port*
  - *Destination Port*
- *UDP*
  - *Source Port*
  - *Destination Port*

and distinct interface.

FFW_RUL_EXT.1.3
The TSF shall allow the following operations to be associated with stateful traffic filtering rules: permit or drop with the capability to log the operation.

FFW_RUL_EXT.1.4
The TSF shall allow the stateful traffic filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.5
The TSF shall:

a.) accept a network packet without further processing of stateful traffic filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [*ICMP*] based on the following *network packet attributes*:
    1. *TCP: source and destination addresses, source and destination ports, sequence number, Flags;*
    2. *UDP: source and destination addresses, source and destination ports;*
    3. *[‘ICMP: source and destination addresses, type, no other protocols]*.

b) Remove existing traffic flows from the set of established traffic flows based on the following: [*session inactivity timeout*].

FFW_RUL_EXT.1.6
The TSF shall enforce the following default stateful traffic filtering rules on all network traffic:

a) The TSF shall drop and be capable of [*counting*] packets which are invalid fragments;
b) The TSF shall drop and be capable of [*counting*] fragmented packets which cannot be re-assembled completely;
c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;
d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network;
e) The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
f) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address “reserved for future use” (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
g) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;
h) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
i) [*no other rules*].

FFW_RUL_EXT.1.7
The TSF shall be capable of dropping and logging according to the following rules:

a) The TSF shall drop and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;
b) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
c) The TSF shall drop and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received.

FFW_RUL_EXT.1.8
The TSF shall process the applicable stateful traffic filtering rules in an administratively defined order.

Application Note: This SFR has been updated as per TD0545.

FFW_RUL_EXT.1.9
The TSF shall deny packet flow if a matching rule is not identified.

FFW_RUL_EXT.1.10
The TSF shall be capable of limiting an administratively defined number of *half-open TCP connections. In the event that the configured limit is reached, new connection attempts shall be dropped and the drop event shall be [counted].*

6.2.5 Identification and Authentication (FIA)

6.2.5.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within *[1-255]* unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

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

Application Note: This SFR has been updated as per TD0570 and TD0571.

6.2.5.2 FIA_PMG_EXT.1 Password Management

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “~”, “,”, “.”, “/”, “:”, “;”, “_”, “+”, “-“, “=”, “{“, “}”, “[“, “]”, “|”, “<”, “>”]
b) Minimum password length shall be configurable to between [*15*] and [*128*] characters.

Application Note: This SFR has been updated as per TD0571.

6.2.5.3 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.

6.2.5.4 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**.

6.2.5.5 FIA_UIA_EXT.1 User Identification and Authentication

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

- Display the warning banner in accordance with FTA_TAB.1;
- [*Respond to ICMP requests*].

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

Application Note: This SFR has been updated as per TD0571.

6.2.5.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates.**
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [*the Online Certificate Status Protocol (OCSP) as specified in RFC 6960*].
- 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.

6.2.5.7 FIA_X509_EXT.2 X.509 Certificate Authentication.

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

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

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

6.2.5.8 FIA_X509_EXT.3 X.509 Certificate Requests

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

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

6.2.6 Security Management (FMT)

6.2.6.1 FMT_MOF.1/Functions Management of Security Functions Behaviour

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

6.2.6.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behavior

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

6.2.6.3 FMT_MOF.1/Services Management of Security Functions Behaviour

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

6.2.6.4 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.*

6.2.6.5 FMT_MTD.1/CryptoKeys Management of TSF Data

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

6.2.6.6 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 [*digital signature*] 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 re-enable an Administrator account;*
    - *Ability to set the time which is used for time-stamps;*
    - *Ability to configure NTP;*
    - *Ability to configure the reference identifier for the peer;*
    - *Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;*
    - *Ability to import X.509v3 certificates to the TOE's trust store;*
    - *Ability to manage the trusted public keys database;*
    - *No other capabilities*].

Application Note: This SFR has been updated as per TD0631.

6.2.6.7 FMT_SMF.1/FFW Specification of Management Functions

FMT_SMF.1/FFW
The TSF shall be capable of performing the following functions:

- *Ability to configure firewall rules;*

6.2.6.8 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.

6.2.7 Protection of the TSF (FPT)

6.2.7.1 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.

6.2.7.2 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.

Application Note: This SFR has been updated as per TD0639.

6.2.7.3 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*].

Application Note: This SFR has been updated as per TD0632.

6.2.7.4 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [*during initial start-up (on power on), periodically during normal operation*] to demonstrate the correct operation of the TSF: [*digital signature verification of the TOE firmware, Entropy health testing, FIPS module self-tests*].

6.2.7.5 FPT_TUD_EXT.1 Trusted Update

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

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

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

6.2.8 TOE Access (FTA)

6.2.8.1 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

6.2.8.2 FTA_SSL.3 TSF-initiated Termination

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

6.2.8.3 FTA_SSL.4 User-initiated Termination

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

6.2.8.4 FTA_TAB.1 Default TOE Access Banners

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

6.2.9 Trusted Path/Channels (FTP)

6.2.9.1 FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1
The TSF shall **be capable of using [*SSH, DTLS*] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [*[DTLS server, DTLS client], 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 (Syslog), SD-WAN communications*].

Application Note: This SFR has been updated as per TD0572.

6.2.9.2 FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin
The TSF shall **be capable of using [*SSH*] 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.*

6.3 TOE SFR Dependencies Rationale for SFRs

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

6.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the PP and any relevant EPs/Modules/Packages, which is/are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 14.

Table 14 – Security Assurance Requirements

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

6.5 Assurance Measures

The TOE satisfied the identified assurance requirements. This section identifies the Assurance Measures applied by Klas Telecom Inc. to satisfy the assurance requirements. The following table lists the details.

Table 15 – TOE Security Assurance Measures

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.

SAR Component	How the SAR will be met
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_CMS.1	
ATE_IND.1	Vendor will provide the TOE for testing.
AVA_VAN.1	Vendor will provide the TOE for testing. Vendor will provide a document identifying the list of software and hardware components.

7 TOE Summary Specification

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

Table 16 – TOE Summary Specification SFR Description

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td>FAU_GEN.1</td><td>The TOE generates a comprehensive set of audit logs that identify specific TOE operation whenever an auditable event occurs. Auditable events are specified in Table 13 – Security Functional Requirements and Auditable Events. Each of the events specified in the audit records is in enough detail to identify the user with which the event is associated, when the event occurred, where the event occurred, the outcome of the event and the type of event that occurred. Administrative tasks of generating, importing and deleting cryptographic keys identify the keys unique name. SSH public keys are identified by the username in the logs on the TOE.</td></tr>
<tr><td>FAU_GEN.2</td><td>The TOE ensures that each auditable event is associated with the identity of the user that triggered the event.</td></tr>
<tr><td>FAU_STG.1</td><td>Audit data is stored locally on the TOE. Data stored locally is kept in an audit log file. Each log file is rotated at approximately 10MB in size but due to the lag between the appending to the log and the rotation of the log, the size may grow larger than this. Each log will never grow larger than 20MB in size. The audit records can’t be deleted by the TOE user or Security Administrator. There are no conditions where a user or administrator can perform a deletion of audit records. The action that the TOE takes when local storage is full is described in FAU_STG_EXT.1.</td></tr>
<tr><td>FAU_STG_EXT.1</td><td>Audit events are stored locally and are also sent to an external audit server in real-time. SSH is used to provide a trusted communication channel with the syslog server. Data stored locally is kept in an audit log file. Each log file is rotated at approximately 10MB in size but due to the lag between the appending to the log and the rotation of the log, the size may grow larger than this. Each log will never grow larger than 20MB in size. The previous log is overwritten by the new log. The TOE user or Security Administrator is not able to modify the audit records. There is a log entry when the file system flash storage is 75% full. The TOE is standalone and audit data is stored locally. Neither a TOE user nor a Security Administrator has system privileges to modify audit records.</td></tr>
<tr><td>FAU_STG_EXT.3/LocSpace</td><td>The TSF generates a log entry when 75% of local flash storage capacity has been used.</td></tr>
<tr><td>FCS_CKM.1</td><td>The TOE supports several cryptographic key generation schemes which include RSA 2048, 3072 and 4096-bit, ECC P-256, ECC P-384, ECC P-521, and FFC safe-prime groups. These are detailed in FCS_CKM.1.
<table>
<tr><th>Key Generation</th><th>SFR</th><th>Usage</th></tr>
<tr><td>RSA</td><td>FCS_DTLSS_EXT.1
FCS_DTLSC_EXT.2
FCS_DTLSS_EXT.1
FCS_DTLSS_EXT.2
FCS_TLSS_EXT.1</td><td>DTLS server and DTLS client.
HTTPS server</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>Elliptic curve</td><td>FCS_SSHS_EXT.1
FCS_SSHC_EXT.1</td><td>SSHS for administration and SSHC tunnel to syslog server</td></tr>
<tr><td>FFC</td><td>FCS_SSHS_EXT.1
FCS_SSHC_EXT.1</td><td>SSHS for administration and SSHC tunnel to syslog server</td></tr>
</table>
</td></tr>
<tr><td>FCS_CKM.2</td><td>In agreement with the key generation schemes the RSA-based, Elliptic curve-based, and Finite field-based key establishment schemes are supported as detailed in FCS_CKM.2.
<table>
<tr><th>Key Establishment Scheme</th><th>SFR</th><th>Usage</th></tr>
<tr><td>RSA</td><td>FCS_DTLSS_EXT.1
FCS_DTLSC_EXT.2
FCS_DTLSS_EXT.1
FCS_DTLSS_EXT.2
FCS_TLSS_EXT.1</td><td>DTLS server and DTLS client.
HTTPS server</td></tr>
<tr><td>Elliptic curve</td><td>FCS_SSHS_EXT.1
FCS_SSHC_EXT.1</td><td>SSHS for administration and SSHC tunnel to syslog server</td></tr>
<tr><td>FFC</td><td>FCS_SSHS_EXT.1
FCS_SSHC_EXT.1</td><td>SSHS for administration and SSHC tunnel to syslog server</td></tr>
</table>
</td></tr>
<tr><td>FCS_CKM.4</td><td>The TOE stores plaintext keys in volatile and non-volatile storage. The TOE satisfies all requirements for destruction of keys and CSPs as specified in FCS_CKM.4. Please refer to Table 19 – Key Storage and Zeroization.</td></tr>
<tr><td>FCS_COP.1/DataEncryption</td><td>The TOE supports AES encryption and decryption conforming to CBC, CTR and GCM as specified in ISO 18033-3 and ISO 10116. The AES key sizes supported are 128 and 256 bits and the AES modes supported are CBC, CTR and GCM. AES is implemented in the following protocols: SSH. Please refer to Table 17 – CAVP Algorithm Certificate References for NIST CAVP certificate numbers for AES.</td></tr>
<tr><td>FCS_COP.1/Hash</td><td>SSH, NTP, and HTTPS support cryptographic hashing using SHA-1, SHA-256, SHA-384, or SHA-512 with message digest sizes of 160, 256, 384, and 512 bits.</td></tr>
<tr><td>FCS_COP.1/KeyedHash</td><td>SSH, NTP, and HTTPS support cryptographic hashing using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, or HMAC-SHA-512 with message digest sizes of 160, 256, 384, and 512 bits. The key length, hash function used, block size, and output MAC lengths are identified in the table below.
<table>
<tr><th>Algorithm</th><th>Block Size</th><th>Key Size</th><th>Digest Size</th></tr>
<tr><td>HMAC-SHA-1</td><td>512 bits</td><td>160 bits</td><td>160 bits</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>HMAC-SHA-256</td><td>512 bits</td><td>256 bits</td><td>256 bits</td></tr>
<tr><td>HMAC-SHA-384</td><td>1024 bits</td><td>384 bits</td><td>384 bits</td></tr>
<tr><td>HMAC-SHA-512</td><td>1024 bits</td><td>512 bits</td><td>512 bits</td></tr>
</table>
</td></tr>
<tr><td>FCS_COP.1/SigGen</td><td>The TOE provides cryptographic signature generation and verification services in accordance with the following cryptographic algorithms:
RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048, 3072, and 4096 bits] according to 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
Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256, 384 or 512 bits] according to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [selection: P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4
EC certificates are not supported for DTLS connections. The Elliptic Curve Digital Signature Algorithm selection only applies to FCS_SSHS_EXT.1 and FCS_SSHC_EXT.1.</td></tr>
<tr><td>FCS_DTLSC_EXT.1 & FCS_DTLSC_EXT.2</td><td>The TOE supports DTLS 1.2 to allow two TOEs to be connected in a SD-WAN and supports both client and server. The following ciphersuites are supported:
TLS_RSA_WITH_AES_128_CBC_SHA;
TLS_RSA_WITH_AES_256_CBC_SHA;
TLS_RSA_WITH_AES_128_CBC_SHA256;
TLS_RSA_WITH_AES_256_CBC_SHA256;
TLS_RSA_WITH_AES_128_GCM_SHA256; and
TLS_RSA_WITH_AES_256_GCM_SHA384.
To initiate a DTLS connection the TOE will send a client hello message. When the hello verify request message is received, the TOE performs a stateless cookie exchange to ensure the DTLS server is not being spoofed. When certificates are exchanged the TOE will confirm that the hostnames match. If the hostnames don't match the DTLS session will not be established.
During internal channel communication between the client and server, if there is a message authentication code (MAC) verification failure, the TOE will silently discard the record and continue with the connection. Key establishment is performed using RSA with 2048 bits, 3072 bits, or 4096 bits.
Valid record sequence numbers are maintained in a sliding window. For each record received, the TOE verifies if it is in the window boundary. Messages that are received where the same record was previously received or that are too old to fit in the sliding window are silently discarded.</td></tr>
</table>

Requirement	TSS Description
	The TOE does not support Elliptic Curves or Group Extensions. Conversion of text representation of IP addresses to binary representation is handled via the netaddr library for python. Canonical format is not enforced for IPv4/IPv6. The TOE supports DTLS mutual authentication and will send its DTLS client-side certificate upon request from a DTLS Server. The TOE supports reference identifiers using FQDN, IPv4 and IPv6 in the CN or SAN of the certificate. Wildcards are not supported for any type of reference identifier. If a SAN and CN are both present in a certificate, SAN takes priority no matter the circumstance.
FCS_DTLSS_EXT.1 & FCS_DTLSS_EXT.2	The TOE supports DTLS 1.2 to allow two TOEs to be connected in a SD-WAN and supports both client and server. The following ciphersuites are supported: • TLS_RSA_WITH_AES_128_CBC_SHA; • TLS_RSA_WITH_AES_256_CBC_SHA; • TLS_RSA_WITH_AES_128_CBC_SHA256; • TLS_RSA_WITH_AES_256_CBC_SHA256; • TLS_RSA_WITH_AES_128_GCM_SHA256; and • TLS_RSA_WITH_AES_256_GCM_SHA384 Upon receiving the client hello message, the TOE sends a hello verify request message and performs a stateless cookie exchange to ensure the DTLS client IP address is not being spoofed. When certificates are exchanged, the TOE will confirm that the FQDN, IPv4 or IPv6 identifier in the CN/SAN matches in the certificate. If the FQDN, IPv4 or IPv6 identifier in the CN/SAN doesn't match, the DTLS session will not be established. If a SAN and CN are both present in a certificate, SAN takes priority no matter the circumstance. During internal channel communication between the client and server, if there is a message authentication code (MAC) verification failure, the TOE will silently discard the record and continue with the connection. Key establishment is performed using RSA with 2048 bits, 3072 bits, or 4096 bits. Parameters are selected based off what key the administrator generates following the guidance documentation. Valid record sequence numbers are maintained in a sliding window. For each record received, the TOE verifies if it is in the window boundary. Messages that are received where the same record was previously received or that are too old to fit in the sliding window are silently discarded. The TOE does not support session resumption with either session ID's or session tickets. Fallback authentication is not supported for DTLS. During Client Authentication only FQDN and IPv4/IPv6 Addresses are supported as identifiers. FQDN input is via an XML defined input field and supports CN-ID, DNS-ID and SRV-ID (per RFC6125), input restrictions prevent application of spaces in text input, characters are limited to (A-Z/a-z/_/-/.). URI-ID format is not supported. IPv4/6 addresses are parsed from an XML input field with restricted input (IPv4:"0-9/.", IPv6:"0-9/A-

Requirement	TSS Description
	F/a-f/:") and matched against expected identifiers (verification that input is compliant with IPv4/IPv6 format) via the netaddr library for python, which receives data from XML input. Within the CN, ip conversion is performed via the inet_ntop function of the standard C++ suite. The TOE supports DTLS mutual authentication and will request the client-side certificate. The TOE does not support fallback authentication for DTLS.
FCS_HTTPS_EXT.1	The TSF uses the RFC 2818 HTTPS protocol for the that complies with RFC 2818. This protocol is used to provide a user with access to a virtual machines (VM) status if a VM is running on the TOE as well as viewing uptime, CPU usage and the time. Peer certificates are not required for authentication. This interface is only used for monitoring functionalities and is not used by an administrator to manage TSF data.
FCS_NTP_EXT.1	The TOE uses NTP v3 (RFC 1305) and uses SHA1 for authenticating time stamps received. The NTP sources are defined by the Security Administrator. Up to three sources can be configured. NTPv3 is implemented on the TOE using Chrony version 3.4.
FCS_RBG_EXT.1	The TOE implements a DRBG in accordance with ISO/IEC 18031:2011 using a CTR DRBG with AES. The noise source is the Intel RDSEED CPU instruction and is seeded with a minimum of 256 bits of entropy. The expected min-entropy rate for the noise source is 0.902120 bits of entropy per bit of noise output.
FCS_SSHC_EXT.1	The TOE implements SSH client that complies with RFC(s) 4251, 4252, 4253, 4254, 5656, and 6668. SSH public key authentication is supported with the following key pairs: ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384. Packet sizes up to 33,292 bytes are accepted and packets exceeding this size are dropped and this event is logged by the TOE. The TOE supports encryption algorithms AES-128-CBC, AES-256-CBC, AES-128-CTR and AES-256-CTR to ensure confidentiality of the session. Password-based authentication is not supported. The TOE supports the following hostkey algorithms: ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384. An IP address is associated with each host-key public key when a key is uploaded to the TOE. The TOE identifies the public key that is presented by the server and verifies if it matches one of the stored keys within the client. If the presented key does not match, authentication is prevented. The TOE supports the following data integrity algorithms: hmac-sha1, hmac-sha2-256, and hmac-sha2-512. The TOE supports the following key exchange algorithms: Diffie-hellman-group14-sha1, ecdh-sha2-nistp256, and ecdh-sha2-nistp384. The TOE supports the following RSA key sizes: 2048, 3072, and 4096. The TOE is capable of rekeying and verifies the following thresholds: • No longer than one hour • No more than 1 GB of transmitted data The TOE continuously checks both conditions. When either of the conditions are met, the TOE will initiate a rekey.

Requirement	TSS Description
FCS_SSHS_EXT.1	The TOE implements SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, 5656, and 6668. SSH password-based authentication and public key authentication are both supported with the following user and host key pairs: ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384. Packet sizes up to 33,292 bytes are accepted and packets exceeding this size are dropped and this event is logged by the TOE. The TOE supports encryption algorithms AES-128-CBC, AES-256-CBC, AES-128-CTR and AES-256-CTR to ensure confidentiality of the session. When a user logs into the TOE, they are authenticated via a username and password or public key. Password-based authentication is not required if a public key is being used. If public key authentication is not available, all users must log in using a password specified for that user account. The password is determined by the user and must conform to the requirements set out in FIA_PMG_EXT.1. When verifying a user's password, the one way hash is computed and the result is checked against the value stored for the username in the /etc/passwd file. Only certain programs on the TOE can access the /etc/passwd file, for example sshd. Users/admins do not have access. The TOE supports the following RSA key sizes: 2048, 3072, and 4096. The TOE identifies the public key that is presented by the client and verifies if it matches one of the stored keys within the server. If the presented key does not match, authentication is prevented. The supported key exchange algorithms are diffie-hellman-group14-sha1, ecdh-sha2-nistp256 and ecdh-sha2-nistp384. The TOE supports the following data integrity algorithms: hmac-sha1, hmac-sha2-256, and hmac-sha2-512 for SSH to ensure integrity of the session. The TOE is capable of rekeying and verifies the following thresholds: • No longer than one hour • No more than 1 GB of transmitted data The TOE continuously checks both conditions. When either of the conditions are met, the TOE will initiate a rekey.
FCS_TLSS_EXT.1	The TOE supports the following TLS_RSA ciphersuites using TLSv1.2: • *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,* • *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,* • *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,* • *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,* • *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,* • *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288* The TOE does not use ECDHE or DHE ciphersuites. Connection attempts for older SSL and TLS versions will be rejected by the TOE. The TOE supports the following RSA key establishment sizes: 2048, 3072 and 4096 bits. The TOE supports session resumption based on session IDs according to RFC 5246. Session resumption is based on a single context and operates according to the applicable RFCs. Sessions can be reused, provided all

Requirement	TSS Description
	session properties and parameters are still valid. If there are any instances where properties are not valid anymore, they are implicitly rejected by the TOE and a full handshake will be performed.
FDP_RIP.2	The TOE ensures that information from previous packets are never transmitted through the TOE. When a packet's memory structure is initially created it is filled with zeroes to ensure that no residual information can be transmitted Packets that are not the required length are padded with zeroes as required before the information is transmitted.
FFW_RUL_EXT.1	When the TOE first boots up, all network interfaces are in a shutdown state until the ACL configuration is processed and loaded. Once the ACL configurations are applied to every interface, then interfaces are enabled and will start processing inbound and outbound packet traffic. This prevents packets from bypassing ACLs during the boot-up process. Stateful traffic filtering is provided for ICMPv4, ICMPv6, IPv4, IPv6, TCP, and UDP network traffic by the traffic filtering service. The TOE can perform packet filtering on all ethernet interfaces. Firewall rules are associated with this distinct network interface type in the same way. The TOE administrator can define rules to permit or drop traffic based on the following parameters: • ICMPv4 Type, Code • ICMPv6: Type, Code • IPv4: Source address, Destination Address, Transport Layer Protocol • IPv6: Source address, Destination Address, Transport Layer Protocol • TCP: Source Port, Destination Port • UDP: Source Port, Destination Port • TOE network interface. The administrator can define whether packets processed by the rules are logged. TCP, UDP, and ICMP packets for established sessions will be allowed without applying the stateful traffic filtering rules based on the parameters defined in FFW_RUL_EXT.1.5. The TOE also has default stateful traffic filtering rules for dropping packets. These rules are defined in FFW_RUL_EXT.1.6. Counting will be performed on these packets. The TOE features a packet filtering capability using stateful Access Control List (ACL) rules configured with the access-list configuration command and interface ip access-group [in|out] settings for IPv4 and IPv6 network traffic. When ACL rules are applied to an interface with the ip access-group [in|out] configuration command, the ACL is either applied to inbound or outbound traffic, depending on the [in|out] option. If an inbound ACL is applied to an interface, inbound packet traffic is checked against the ACL rules applied to that interface, and either allowed or dropped depending on the configuration of the matching rule before any further processing of the packet occurs (such as routing, etc.). If an outbound ACL is applied to

Requirement	TSS Description
	an interface, then any packets that are queued to be sent out an interface are checked against the configured ACL and either allowed to be sent or dropped at the outbound interface. The TOE stores ACL rules in the order they were configured. This is the same order the rules are checked when checking an inbound or outbound packet. Whenever any ACL is applied to an interface, a default DROP policy rule is applied for that interface as the last rule. This will drop any packet that didn't match any other rule in the applied ACL. For example, if an ACL is applied in the outbound direction, then by default, an outbound packet that didn't match any rules will be dropped by default. If there are conflicting rules configured, the first rule in the list will be processed. Packets that are allowed through any inbound ACL are then checked against the stateful session table to see if there is an existing session to which the packet belongs. Packet information such as source and destination IP address, source and destination ports, protocol, and flags unique to protocols are used to determine if the packet belongs to an existing session or not. If no existing session matches the packet, a new session is created. The TOE will keep track of stateful sessions in the table until either the protocol ends the session (such as TCP-FIN or TCP-RST packets) or after an amount of time has lapsed (timeout period) where no packet was matched against the session. The exact timeout period depends on the session type and current state of the session and is not configurable. Examples: ICMP and ICMPv6 is 30 seconds, TCP sessions in the FIN WAIT state is 120 seconds, etc. Half-open TCP connection attacks are mitigated with a synproxy option feature which can be configured with an ACL rule. This option causes the TOE to intercept new TCP connections and determines if the packet is a false SYN-ACK or ACK packet that should be dropped. Specific ports can be configured with the ACL rule as usual. This feature is either on or off and does not permit the configuration of a number of half-open TCP states allowed. The dropping of these packets is logged. The default threshold limit for half open connections on the TOE is 0. In all cases, TCP half open connections are monitored per TCP client. The default state is to allow 2048 half open TCP connections if nothing is configured. After this number is reached, new SYN packets are dropped. Each new SYN attempt is monitored for 31 seconds, based on retrying 5 SYN-ACK responses. After 31 seconds, the stale half open connections are removed. Whenever packet traffic exceeds the maximum rate the TOE can handle, the TOE drops the excess traffic. This ensures that traffic which cannot be processed but does not match firewall filter rules will not be passed through. There is an integrated firewall ruleset that can be applied when the packets have a source address equal to the address of the network interface where the packet was received. It must be configured for IPv4 and IPv6 separately. This configuration allows the security administrator to specify valid IP address ranges for the source address of incoming packets. At the end of the firewall ruleset, there is a default-deny that will reject any packets with an invalid source address. This rule will also deny any

Requirement	TSS Description

	traffic where the source address does not belong to the networks associated with the network interface where the packet was received.
	
	Components involved in processing network packets:
	• Network card
	• KlasOS operating system on x86 processor
	  ○ Boot time failure (cryptographic initialization, other)
	  ○ Run time failure
	  ▪ Drop network packets due to slow processing
	
	In the event of failure of the network card, packets may not be processed.
	
	<u>Boot for KlasOS</u>
	
	Interfaces are initialized to DOWN state until the firewall configuration is fully applied. If a boot time failure is discovered, for example failure to apply a configuration due to an internal error, the interfaces remain DOWN, causing no traffic to flow.
	
	If there is a cryptographic initialization error, e.g. an algorithm check fails, the same thing applies.
	
	<u>Run time packet processing</u>
	
	Once KlasOS Firewall is fully configured, the kernel of KlasOS is used as the firewall component. In the event of kernel failure, the device shuts down. However, this is extremely unlikely to occur.
	
	If packets arrive at a higher rate than the ability of KlasOS to process them, the excess packets are dropped.
	
	Firewall rules exist to allow preferential treatment of established (stateful) TCP/IP and UDP flows. If these rules are configured, it is more likely for firewall selected 'good' flows to remain connected.
	
	Secondly, the firewall has rules to rate limit and 'synproxy' flows which have not yet become established.
	
	When properly configured the Firewall mitigates DDOS and SYN flood attacks, so that resources are allocated for 'good' user traffic.
	
	There is also an integrated firewall ruleset that can be applied when the source or destination address of the network packet is a link-local address.

FIA_AFL.1	An administrator can configure the maximum number of failed attempts using the CLI interface. The configurable range is between 1 and 255 attempts. When a user account has sequentially failed authentication for the configured number of times, the account will be locked, until a local administrator manually unlocks the account. If the lockout attempts are set to, for example, 5 attempts, then the user will be locked out after the 5th consecutive failed login attempt. This means that the 6th and subsequent attempts will fail to gain access to the TOE even if the credential being offered is correct. All failed attempts and lockouts are tracked by the TOE audit logs. The TOE will always allow a user to authenticate using the local console port, even if the user account is locked. This behavior is not configurable.

Requirement	TSS Description
FIA_PMG_EXT.1	The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of upper- and lower-case letters, numbers, and at least one of the special characters that include the following: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “~”, “<”, “>”, “,”, “.”, “/”, “:”, “;”, “_”, “+”, “-“, “=”, “{“, “}”, “[“, “]”, “|”. The minimum password length can be configured by the Administrator and can range from 15 to 128 characters.
FIA_UAU.7	For all authentication at the local CLI the TOE does not display any authentication data. The TOE does not display even obscured data such as asterisks during authentication attempts.
FIA_UIA_EXT.1	The TOE requires all users to be successfully identified and authenticated as an administrator before allowing any TSF mediated actions to be performed. Access to the TOE is facilitated through by directly connecting to the TOE through serial console or remotely connecting to the TOE through SSHv2. Every user that authenticates is first logged in with non-administrative privileges with limited viewing functionalities. The user may then authenticate as an administrator with additional credentials to gain access to modifying functionalities. Regardless of the interface at which the administrator interacts, the TOE prompts the user for a credential. Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. No TOE administrative access is permitted until an administrator is successfully identified and authenticated. For remote administration, the TOE supports public key authentication and password-based authentication. If the user uses public key-based authentication and it is successful, then the user is granted access to the TOE. If the user uses password-based authentication and they provide valid username and password, then user is granted access to the TOE. If the user enters invalid user credentials, they will not be granted access. The TOE does not provide a reason for failure in case of a login failure. The TOE displays a banner in accordance with FTA_TAB.1 before a user can log into the device. The TOE responds to ICMP requests without prior authentication.
FIA_UAU_EXT.2	The TOE uses local password-based authentication.
FIA_X509_EXT.1/Rev	The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for secure DTLS connections. Certificates are used to authenticate and establish secure SDWAN communication. The TOE will check the validity of the DTLS Server certificate prior to establishing a secure DTLS connection. The certificate validation is determined based on reference ID verification, certificate path, extended key usage field, certificate expiry date, and the certificate revocation status. The TOE also validates certificates in accordance with the following rules: • RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.

Requirement	TSS Description
	• The certification path must terminate with a trusted CA certificate designated as a trust anchor. • The TOE validates 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 TOE validates the revocation status of the certificate using Online Certificate Status Protocol (OCSP) as specified in RFC 6960. • The TOE validates the extendedKeyUsage field according to the following rules: o Server certificates presented for DTLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field o Client certificates presented for DTLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field. o 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 extended key usage field. For an expired certificate, TOE will deny the connection. During secure SDWAN connection establishment, any byte modification in the certificate will lead to connection failure. The TOE used OCSP for revocation checking. If the validation of the certificate fails because the OCSP Server cannot be connected to it, the certificate shall not be accepted, and the connection is terminated. It verifies whether the certificate or intermediate CA certificate has been revoked when a leaf certificate is presented to the TOE as part of the certificate chain during authentication. Revocation checking is done during authentication on all certificates in chain using OCSP. If only a leaf is presented to the TOE, that certificate will also be checked for its revocation status. The TOE only supports RSA-based certificates. The TOE relies on CA certificates that are imported in its trust store to be able to validate and make connections with the IT entity. The TOE only sends leaf certificates as part of the connection and validates the presented chain against the CA certificates stored in its trust store.
FIA_X509_EXT.2	Certificates to support DTLS can be configured from the CLI and SSH interfaces. If the TSF determines that the certificate is not valid when the DTLS channel is being setup, it will not accept the certificate. If a connection cannot be established during the validity check of a certificate, the TOE will not accept the certificate and application data will not flow.
FIA_X509_EXT.3	When generating a certificate request the TSF provides the public key and common name in the request. Device-specific information is not provided as part of the CSR.

Requirement	TSS Description
FMT_MOF.1/Functions	The Security administrator can configure a SSH tunnel for secure transmission of audit data to a syslog server. The IP address of the system log and the port to be used can be configured.
FMT_MOF.1/ManualUpdate	The TOE restricts the ability to perform software updates to Security Administrators.

FMT_MOF.1/Services	The TOE may be managed via the CLI (console and remote SSH). The specific services the administrator can start and stop and how they do it are shown below:
	• SSH Administration
	  ○ Enabling and disabling remote SSH access can be done via the CLI
	• SSH syslog connections
	  ○ Enabling and disabling SSH syslog can be done via the CLI
	• SD-WAN Connections
	  ○ Enabling and disabling SD-WAN can be done via the CLI
	• HTTPS limited GUI
	  ○ Enabling and disabling the GUI can be done via the CLI
	Local console and remote administration provide the same functionalities based on the level of authentication.

FMT_MTD.1/CoreData	The TOE restricts the ability to manage the TOE to Security Administrators. Administrative users are required to login before being provided with access to any administrative functions. Non-security administrators are not allowed to modify any TOE functions. No interface is available to an unauthenticated user except the login prompt. Any commands used to modify TOE functions are not made available to non-administrative users and its attempt to use them will result in an invalid action error. The ability to modify the TOE's trust store (modify, import, generate) X509 certificates is restricted to the security administrator.
FMT_MTD.1/CryptoKeys	The security administrator can generate, import, and delete cryptographic keys through the TOE's Global Configuration mode. The specific keys they can manage are listed below: • SSH public keys used for FCS_SSHS_EXT.1 and FCS_SSHC_EXT.1 • X509 Public keys used for FCS_DTLSS_EXT.1 and FCS_DTLSC_EXT.1 • Certificates used for DTLS connections • RSA keys used for HTTPS connections under FCS_TLSS_EXT.1
FMT_SMF.1	The available management functions are listed below and these can be accessed via the SSH command line interface remotely. The local interface can be accessed via a serial port and is identified with "tty" in the audit record. • *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;*

Requirement	TSS Description

	• *Ability to update the TOE, and to verify the updates using [*digital signature*] capability prior to installing those updates;*
	• *Ability to configure the authentication failure parameters for FIA_AFL.1;*
	• [
	  ○ *Ability to 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 re-enable an Administrator account;*
	  ○ *Ability to set the time which is used for time-stamps;*
	  ○ *Ability to configure NTP;*
	  ○ *Ability to configure the reference identifier for the peer;*
	  ○ *Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;*
	  ○ *Ability to import X.509v3 certificates to the TOE's trust store;*
	  ○ *Ability to manage the trusted public keys database;*
	  ○ *No other capabilities*].

FMT_SMF.1/FFW	The administrator can configure firewall rules via the command line interface both locally and remotely.
FMT_SMR.2	The TOE supports a security administrator role. The security administrator can administer the TOE locally or remotely. The TOE also supports a user role when logging into the remote web GUI via HTTPS. This interface does not have any administrator privileges and the user is not allowed to make any configuration changes to the TOE.
FPT_APW_EXT.1	The TOE stores all password authentication data in a secure directory that is not readily accessible to administrators. Passwords are obscured from the user from both local and remote CLI interfaces. The passwords are stored as SHA-512 hash and are not in plaintext.
FPT_SKP_EXT.1	The TOE stores all private, symmetric and asymmetric keys in secure storage and is not accessible through an interface to administrators. Passwords are obscured from the user from local and remote CLI interfaces. The TOE stores all password authentication data in a secure directory that is not accessible to administrators. Private keys may be destroyed or replaced but cannot be read. Refer to Table 19 – Key Storage and Zeroization for key storage for more details.
FPT_STM_EXT.1	The TOE provides reliable time stamps. The clock function is reliant on the system clock provided by the underlying hardware. This clock is kept accurate and reliable using NTP. The following security functions make use of the system time: • Audit events • Session inactivity • SSH Rekey The time can be manually updated by a Security Administrator.

Requirement	TSS Description
FPT_TST_EXT.1	The TOE runs a suite of self-tests during initial start-up to verify its correct operation. If any of the tests fail, the TOE will enter an error state. The TOE executes the following self-tests when powered on: • Integrity check – The TOE performs an integrity check of the installed firmware by comparing the 4096-bit digital signature of the complete firmware image during bootup before any configuration is loaded and interfaces are enabled. If signature verification fails, all SSH functionality is disabled and a message will be sent to the system log. • FIPS module self-tests in accordance with the OpenSSL 3.0.8 FIPS 140-2 Policy – The TOE performs FIPS self-tests to test the integrity of the operational environment when the cryptographic module is first initialized during boot-up. This includes KAT and PCT on all supported algorithms. If any cryptographic self-test fails, the TOE will complete the boot process with all cryptographic functions disabled. The entropy noise source health tests are performed during bootup as part of the self-tests. They also are run continuously during system runtime. • Entropy health testing – If the entropy noise source health testing fails, the TOE immediately reboots and logs an audit message at the local console.
FPT_TUD_EXT.1	Before posting a new image for customer download, Klas creates a SHA256 hash of the image and then cryptographically digitally signs the hash using an RSA private key. This signed hash is then appended to the end of the firmware image. The public key is burned into the image already. The key that is burned into the image is used to validate the cryptographic signature of the update file. The Security Administrator can query the software version running on the TOE using the 'show version' command and is able to perform manual software updates. When software updates are made available by Klas, the Security Administrator can download and initiate installation of the update. The TOE will verify that the signed hash on the new image is valid before booting with the new image. If the image fails the signature check, then the image is deleted from the device and no upgrade occurs.
FTA_SSL.3 FTA_SSL_EXT.1	A Security Administrator can configure maximum inactivity times for administrative sessions through the TOE, local CLI, and remote SSH interfaces. The configuration of inactivity periods are applied on a per-interface basis and can be applied to both local, and remote sessions in the same manner. When the interface has been idle for more than the configured period of time, the session will be terminated and will require reauthentication to establish a new session.
FTA_SSL.4	A Security Administrator is able to exit out of both local, and remote administrative sessions. For both local and remote sessions, the session is terminated by entering the "exit" command.
FTA_TAB.1	Access to the TOE is facilitated through by directly connecting to the TOE through serial console or remotely connecting to the TOE through SSHv2. Security Administrators can define a customized login banner that will be

Requirement	TSS Description
	displayed at the local CLI and remote CLI (SSH). This banner will be displayed prior to allowing Security Administrators access.
FTP_ITC.1	A remote audit server can be configured and the communication between the TOE and the audit server is protected by SSHv2 tunnel using public-key based authentication. The TOE acts as a client in the syslog connection. One or more TOEs may be connected in a SD-WAN and these connections are protected by DTLS. Though TSF data is not transmitted between TOEs, this SD-WAN connection could be used to administer another TOE. In this case the administrator session would be protected by the DTLS SD-WAN connection and SSH. The TOE can act as both a client and server in the SD-WAN connections. All cryptographic information that pertains to syslog connections can be found under FCS_SSHC_EXT.1. All cryptographic information that pertains to SD-WAN connections can be found under FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2, FCS_DTLSC_EXT.1 and FCS_DTLSC_EXT.2.
FTP_TRP.1/Admin	Remote administration is performed using a CLI interface that is protected by SSHv2 using AES encryption. All requirements that secure this connection can be found in FCS_SSHS_EXT.1.

7.1 CAVP Algorithm Certificate Details

The TOE uses OpenSSL 3.0.8 and the associated algorithms are presented in the table below. The CAVP certificate is A4573.

Table 18 – CAVP Algorithm Certificate References

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3	KlasOS Keel	RSA KeyGen	#A4573
	ECC schemes using “NIST curves” [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4	KlasOS Keel	ECDSA KeyGen	#A4573
	FFC Schemes using ‘safe-prime’ groups that meet the following: “NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment	KlasOS Keel	Safe-Primes key generation Safe-Primes Key Verification	#A4573

	Schemes Using Discrete Logarithm Cryptography" and [RFC 3526, RFC 7919].			
FCS_CKM.2	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"	KlasOS Keel	None: CCTL tested as per the PP/SD Evaluation Activities	Tested with known-good implementation
	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	KlasOS Keel	KAS-ECC-SSC	#A4573
	FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 3526].	KlasOS Keel	KAS-FFC-SSC	#A4573
FCS_COP.1/ DataEncryption	AES used in [CBC, CTR, and GCM] mode and cryptographic key sizes [128 bits, 256 bits]	KlasOS Keel	AES-CBC 128 bits, 256 bits AES-GCM 128 bits, 256 bits AES-CTR 128 bits, 256 bits	#A4573

FCS_COP.1/ SigGen	For RSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	KlasOS Keel	RSA-SigGen RSA-SigVer 2048, 3072 and 4096	#A4573
	For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4	KlasOS Keel	ECDSA-SigGen ECDSA-SigVer P-256, P-384, P-521	#A4573
FCS_COP.1/ Hash	[SHA-1, SHA-256, SHA-384, SHA-512] and message digest sizes [160, 256, 384, 512] bits	KlasOS Keel	SHA-1 SHA2-256 SHA2-384 SHA2-512	#A4573
FCS_COP.1/ KeyedHash	[HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes [160, 256, 384, and 512 bits] and message digest sizes [160, 256, 384, 512] bits	KlasOS Keel	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512	#A4573
FCS_RBG_EXT.1	CTR_DRBG (AES-256)	KlasOS Keel	Counter DRBG AES 256	#A4573

7.2 Cryptographic Key Destruction

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

Table 19 – Key Storage and Zeroization

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
EC Session Keys	Ephermeral Session Key for SSH Session Establishment	Epheremal; stored in RAM (Volatile storage)	Overwritten with zeroes at end of session.
Diffie Hellman Group 14 Session Keys	Ephermeral Session Key for SSH Session Establishment	Epheremal; stored in RAM (Volatile storage)	Overwritten with zeroes at end of session.
RSA Key	Signature Generation, Signature Verification for SSH public key authentication.	Restricted key partition in plaintext (Non-Volatile storage)	Deleted with read-verify when any of the designated cryptographic key zeroization commands identified in AGD are executed by the administrator. Key zeroization will instruct a part of the TOE to destroy the abstraction that represents the key. Generating a new key will overwrite and erase any existing keys and replacing the old keys with a new key value.
		While in use, RSA keys are held in RAM (Volatile storage)	Overwritten with zeroes when the key is no longer in use (after performing a cryptographic operation) or overwritten with a new value of the key when a new key value.
ECDSA Key	Signature Generation. Signature Verification for SSH public key authentication and verification of trusted updates.	Restricted key partition in plaintext (Non-Volatile storage)	Deleted with read-verify when any of the designated cryptographic key zeroization commands identified in AGD are executed by the administrator. Key zeroization will instruct a part of the TOE to destroy the abstraction that represents the key. Generating a new key will overwrite and erase any existing keys and replacing the old keys with a new key value.

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
		While in use, ECDSA keys are held in RAM (Volatile storage)	Overwritten with zeroes when the key is no longer in use (after performing a cryptographic operation)
HMAC Key	Keyed Hashing for SSH	While in use, keys for HMAC keyed hashing are held in RAM (Volatile storage)	Overwritten with zeroes when the key is no longer in use (after performing a cryptographic operation).
AES Session Keys	SSH Data Encryption	Ephemeral; stored in RAM (Volatile storage)	Overwritten with zeroes at end of session

8 Acronym Table

Table 20 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CLI	Command Line Interface
CTR_DRBG	A random number generator
DSS	Digital Signature Standard
DTLS	Datagram Transport Layer Security
EP	Extended Package
FFC	Finite Field Cryptography
HMAC	Hash-based Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
IEC	International Electrotechnical Commission
IP	Internet Protocol
ISO	International Organization for Standardization
NDcPP	Network Device Collaborative Protection Profile
NIAP	Nation Information Assurance Partnership
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
PP	Protection Profile
RBG	Random Bit Generator
RFC	Technical specification documents
RSA	Rivest, Shamir & Adleman
SD-WAN	Software Defined Wide Area Network
SFR	Security Functional Requirement
SHA	Secure Hash
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSF	TOE Security Functionality
TSS	TOE Summary Specification