SpaceX Regulus Security Target

Document Version: 1.2

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History

Version	Date	Changes
Version 1.0	May 2023	Initial Release
Version 1.1	July 2023	Updates based on ECR comments.
Version 1.2	August 2023	Updates based on 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	SpaceX Regulus Security Target
ST Version	1.2
ST Date	August 2023
ST Author	Acumen Security, LLC.
TOE Identifier	SpaceX Regulus
TOE Version	1.0
TOE Hardware	Apogee-100
TOE Developer	Space Exploration Technologies Corp.
Key Words	Network Device, VPN Gateway

1.2 TOE Overview

The SpaceX Regulus TOE is classified as a VPN Gateway, which is a Network Device composed of both hardware and software that is connected to networks and provides IPsec protection of network traffic. The SpaceX Regulus TOE is comprised of the Apogee-100 hardware running firmware version 1.0.

1.3 TOE Description

This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references. Below is a diagram of the representative TOE deployment in its evaluated configuration:

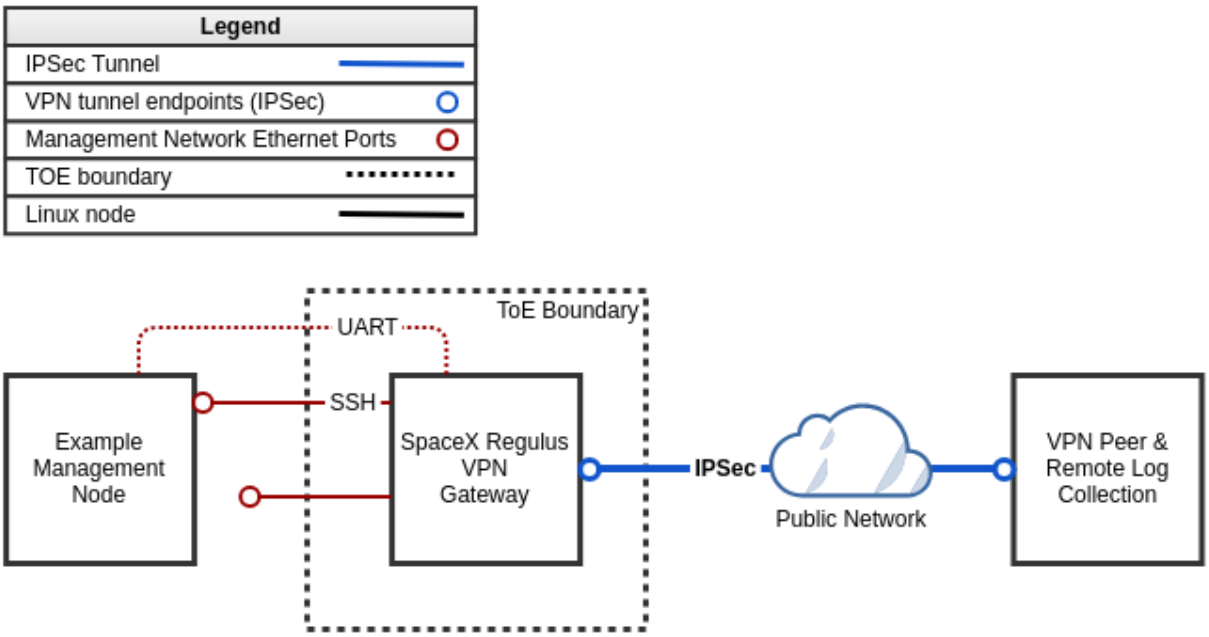

Figure 1 – Representative TOE Deployment

1.3.1 Physical Boundaries

The physical boundary of the TOE is the SpaceX Regulus chassis, which is a networked device providing connectivity to external networked entities. The TOE includes a specialized PCB board containing a Zynq Ultrascale+ ZU5 System on Chip (SoC) processor, based on Armv8-A Architecture, which executes the TOE software along with a NXP SE050F cryptographic accelerator (Secure Element Chip). The TOE provides the following interfaces for management and network connectivity:

- 1x 100Mbps and 1x 10Gbps Ethernet ports for connectivity to trusted networks
- 1x 100Mbps, 1x 1Gbps, and 1x 10Gbps Ethernet ports for connectivity to untrusted networks
- UART for local serial console access
- 120VAC power input

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, along with the functionality specified in the PP-Module for VPN Gateways, or MOD_VPNGW 1.1.

1.3.2.1 Security Audit

The TOE generates audit events for all actions specified in Table 11 below and includes the identity of the entity that caused the event (if applicable), date and time of the event, event type, and outcome. Audit records are transmitted to an external log receiver via IPsec tunnels.

1.3.2.2 Cryptographic Support

The TOE implements CAVP validated cryptographic algorithms as specified in section 6.1 for asymmetric key generation, encryption/decryption, digital signatures, hashing, message authentication, and random bit generation. These algorithms are used to provide security for the SSH and IPsec connections, DRBG Operations, secure key generation and storage, digital signature operations, IPsec and SSH algorithm support, and digital signature operations. IPsec

1.3.2.3 Identification and Authentication

Identification and authentication are required both for user administrative access to the device and for establishing IPsec VPN peer connections.

User-level authentication is performed at the command line and supports remote and local access with public key authentication and passwords for SSH over the network and password authentication only for local console access. No management functionality is granted to users prior to this authentication process and all trusted passwords and SSH keys are stored locally on the TOE. Passwords must be a minimum length of 15 characters and only ECDSA P-384 keys are supported for pubkey authentication. If a user fails to authenticate via a password, their account will be automatically locked to remote access until an administrator-configurable amount of time has passed.

Authentication with an IPsec VPN peer is first established with IKEv2 based on X.509 ECDSA certificates. Peers that attempt to authenticate using certificates that are specified via CRLs will be rejected during the key exchange process. IPsec tunnels will not be established until the IKE process has been completed successfully for the full chain of trust.

1.3.2.4 Security Management

The security management functionality including access to cryptographic keys and TSF data is limited to the Security Administrator role. The TOE is managed via a remote SSH CLI and local serial CLI.

1.3.2.5 Packet Filtering

The TOE provides packet filtering and secure IPsec tunneling between the TOE and a trusted VPN endpoint.

1.3.2.6 Protection of the TOE Security Functionality (TSF)

The TOE prevents the reading of secret keys, private keys and passwords. During initial startup, the TOE runs a suite of self-tests to demonstrate correct operation of the cryptographic functionality. The TOE provides a means to verify firmware/software updates to the TOE using digital signature prior to installing those updates. The TOE provides reliable time stamps for itself.

1.3.2.7 TOE Access

The TOE terminates inactive remote and local sessions after an administrator configurable time-period. Sessions can also be terminated by the administrative user. The TOE also displays a configurable login banner prior to authenticating the user.

1.3.2.8 Trusted Path/Channels

The TOE provides a trusted path for administration via SSH. Trusted channels are implemented via IPsec to VPN endpoints as well as for audit log receivers.

1.3.3 TOE Documentation

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

- SpaceX Regulus Security Target (this document)
- Common Criteria Configuration Guide for Regulus VPN with SpaceX OS v1.0; version 0.1; March 2023

1.4 TOE Environment

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

Table 2 – Required Environmental Components

Component	Purpose/Description
VPN Peer	Peer VPN endpoint and audit log receiver
Management PC	Local/remote management
TFTP Server	The TFTP server hosts TOE update files and transfers them to the TOE. However, the TFTP service itself is out of scope of the evaluation.

2 Conformance Claims

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

2.1 CC Conformance Claims

The TOE is conformant to the following:

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

2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- PP-Configuration for Network Device and Virtual Private Network (VPN) Gateways, Version 1.1, 01 July 2020

  This PP-Configuration includes the following:
  - collaborative Protection Profile for Network Devices, Version 2.2e [CPP_ND_V2.2E]
  - PP-Module: PP-Module for Virtual Private Network (VPN) Gateways, Version 1.1 [MOD_VPNGW_V1.1]

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) and Module, performing only the operations defined there.

2.3.1 Technical Decisions

All NIAP TDs issued to date and applicable to NDcPP v2.2e and the MOD_VPNGW v1.1 have been considered. Table 3 identifies all applicable TDs.

Table 3 – Relevant Technical Decisions

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0527: Updated to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	No	NTP is not supported by the TOE.
TD0536: NIT Technical Decision for Update Verification Inconsistency	Yes	
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	No	TLS is not supported by the TOE.
TD0546: NIT Technical Decision for DTLS – clarification of Application Note 63	No	DTLS is not supported by the TOE.

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
TD0549: Consistency of Security Problem Definition update for MOD_VPNGW_v1.0 and MOD_VPNGW_v1.1	Yes	
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	TLS is not supported by the TOE.
TD0556: NIT Technical Decisions for RFC 5077 question	No	TLS is not supported by the TOE.
TD0563: NIT Technical Decision for Clarification of audit date information	Yes	
TD0564: NIT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	DTLS is not supported by the TOE.
TD0570: NIT Technical Decision for Clarification about FIA_AFL.1	Yes	
TD0571: NIT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
TD0572: NIT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Yes	
TD0590: Mapping of operational environment objectives	Yes	
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	No	The TOE is not virtual.

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0592: NIT Technical Decision for Local Storage of Audit Records	Yes	
TD0597: VPN GW IPv6 Protocol Support	Yes	
TD0631: NIT Technical Decision for Clarification of public key authentication for SSH server	Yes	
TD0632: NIT Technical Decision for Consistency with Time Data for vNDs	No	The TOE is not virtual.
TD0633: NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Yes	
TD0634: NIT Technical Decision for Clarification required for testing IPv6	No	TLS/DTLS is not supported by the TOE.
TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters	No	TLS is not supported by the TOE.
TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	SSH client is not supported by the TOE.
TD0638: NIT Technical Decision for Key Pair Generation for Authentication	Yes	
TD0639: NIT technical decision for clarification for NTP MAC keys	No	The TOE does not implement NTP or FCS_NTP.
TD0670: NIT Technical decision for mutual and non-mutual auth TLSC testing	No	The TOE does not implement TLSC
TD0738 - NIT Technical Decision for Link to Allowed-With List	Yes	

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 4 are drawn directly from the PP and any EPs/Modules/Packages specified in Section 2.2.

Table 4 – Threats

ID	Threat
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of

ID	Threat
	confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic and may allow them to take advantage of any trust relationships with other Network Devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.
T.DATA INTEGRITY	Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to modify the data without authorization. If known malicious external devices are able to communicate with devices on the protected network or if devices on the protected network can communicate with those external devices then the data contained within the communications may be susceptible to a loss of integrity.
T.NETWORK_ACCESS	Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network. Devices located outside the protected network may, likewise,

ID	Threat
	offer services that are inappropriate for access from within the protected network. From an ingress perspective, VPN gateways can be configured so that only those network servers intended for external consumption by entities operating on a trusted network (e.g., machines operating on a network where the peer VPN gateways are supporting the connection) are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network. From an egress perspective, VPN gateways can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network, or moreover are accessed via an encrypted channel. For example, access to external mail services can be blocked to enforce corporate policies against accessing uncontrolled e-mail servers, or, that access to the mail server must be done over an encrypted link.
T.NETWORK_DISCLOSURE	Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to conduct unauthorized activities. If known malicious external devices are able to communicate with devices on the protected network, or if devices on the protected network can establish communications with those external devices (e.g., as a result of a phishing episode or by inadvertent responses to email messages), then those internal devices may be susceptible to the unauthorized disclosure of information. From an infiltration perspective, VPN gateways serve not only to limit access to only specific destination network addresses and ports within a protected network, but whether network traffic will be encrypted or transmitted in plaintext. With these limits, general network port scanning can be prevented from reaching protected networks or machines, and access to information on a protected network can be limited to that obtainable from specifically configured ports on identified network nodes (e.g., web pages from a designated corporate web server). Additionally, access can be limited to only specific source addresses and ports so that specific networks or network nodes can be blocked from accessing a protected network thereby further limiting the potential disclosure of information. From an exfiltration perspective, VPN gateways serve to limit how network nodes operating on a protected network can connect to and communicate with other

ID	Threat
	networks limiting how and where they can disseminate information. Specific external networks can be blocked altogether or egress could be limited to specific addresses and/or ports. Alternately, egress options available to network nodes on a protected network can be carefully managed in order to, for example, ensure that outgoing connections are encrypted to further mitigate inappropriate disclosure of data through packet sniffing.
T.NETWORK_MISUSE	Devices located outside the protected network, while permitted to access particular public services offered inside the protected network, may attempt to conduct inappropriate activities while communicating with those allowed public services. Certain services offered from within a protected network may also represent a risk when accessed from outside the protected network. From an ingress perspective, it is generally assumed that entities operating on external networks are not bound by the use policies for a given protected network. Nonetheless, VPN gateways can log policy violations that might indicate violation of publicized usage statements for publicly available services. From an egress perspective, VPN gateways can be configured to help enforce and monitor protected network use policies. As explained in the other threats, a VPN gateway can serve to limit dissemination of data, access to external servers, and even disruption of services – all of these could be related to the use policies of a protected network and as such are subject in some regards to enforcement. Additionally, VPN gateways can be configured to log network usages that cross between protected and external networks and as a result can serve to identify potential usage policy violations.
T.REPLAY_ATTACK	If an unauthorized individual successfully gains access to the system, the adversary may have the opportunity to conduct a “replay” attack. This method of attack allows the individual to capture packets traversing throughout the network and send the packets at a later time, possibly unknown by the intended receiver. Traffic is subject to replay if it meets the following conditions: ● Cleartext: an attacker with the ability to view unencrypted traffic can identify an appropriate segment of the communications to replay as well in order to cause the desired outcome. ● No integrity: alongside cleartext traffic, an attacker can make arbitrary modifications to captured traffic and replay it to cause the desired outcome if the recipient has no means to detect these.

3.2 Assumptions

The assumptions included in Table 5 are drawn directly from PP and any relevant EPs/Modules/Packages.

Table 5 – Assumptions

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

ID	Assumption
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.COMPONENTS_RUNNING (applies to distributed TOEs only)	For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.
A.CONNECTIONS	It is assumed that the TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

3.3 Organizational Security Policies

The OSPs included in Table 6 are drawn directly from the PP and any relevant EPs/Modules/Packages.

Table 6 – 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 7 – Security Objectives

ID	Security Objectives
O.ADDRESS_FILTERING	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption or denial of services, and network-based reconnaissance, compliant TOE's will implement Packet Filtering capability. That capability will restrict the flow of network traffic between protected networks and other attached networks based on network addresses of the network nodes originating (source) and/or receiving (destination) applicable network traffic as well as on established connection information. Addressed by: FPF_RUL_EXT.1, FTA_VCM_EXT.1 (optional)
O.AUTHENTICATION	To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (IPSec) will allow a VPN peer to establish VPN connectivity with another VPN peer and ensure that any such connection attempt is both authenticated and authorized. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity. Addressed by: FCS_IPSEC_EXT.1 (refined from Base-PP), FIA_X509_EXT.1/Rev (from Base-PP), FIA_X509_EXT.2 (refined from Base-PP), FIA_X509_EXT.3 (from Base-PP), FTP_ITC.1/VPN, FTA_SSL.3/VPN (optional), FTA_TSE.1 (optional), FIA_PSK_EXT.1 (selection-based)
O.CRYPTOGRAPHIC_FUNCTIONS	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption of services, and network-based reconnaissance, compliant TOE's will implement a cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE. Addressed by: FCS_COP.1/DataEncryption (refined from Base-PP), FCS_IPSEC_EXT.1 (refined from Base-PP), FCS_CKM.1/IKE, FIA_PSK_EXT.1 (selection-based)
O.FAIL_SECURE	There may be instances where the TOE's hardware malfunctions or the integrity of the TOE's software is compromised, the latter being due to malicious or

ID	Security Objectives
	non-malicious intent. To address the concern of the TOE operating outside of its hardware or software specification, the TOE will shut down upon discovery of a problem reported via the self-test mechanism and provide signature-based validation of updates to the TSF. Addressed by: FPT_TST_EXT.1 (refined from Base-PP), FPT_TUD_EXT.1 (refined from Base-PP), FPT_FLS.1/SelfTest, FPT_TST_EXT.3
O.PORT_FILTERING	To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) and/or receiving (destination) port (or service) identified in the network traffic as well as on established connection information. Addressed by: FPF_RUL_EXT.1
O.SYSTEM_MONITORING	To address the issues of administrators being able to monitor the operations of the VPN gateway, it is necessary to provide a capability to monitor system activity. Compliant TOEs will implement the ability to log the flow of network traffic. Specifically, the TOE will provide the means for administrators to configure packet filtering rules to 'log' when network traffic is found to match the configured rule. As a result, matching a rule configured to 'log' will result in informative event logs whenever a match occurs. In addition, the establishment of security associations (SAs) is auditable, not only between peer VPN gateways, but also with certification authorities (CAs). Addressed by: FAU_GEN.1 (refined from Base-PP), FPF_RUL_EXT.1
O.TOE_ADMINISTRATION	TOEs will provide the functions necessary for an administrator to configure the packet filtering rules, as well as the cryptographic aspects of the IPsec protocol that are enforced by the TOE. Addressed by: FMT_MTD.1/CryptoKeys (refined from Base-PP), FMT_SMF.1 (refined from Base-PP) FMT_SMF.1/VPN

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 8 – Security Objectives for the Operational Environment

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

5 Security Requirements

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

Table 9 – SFRs

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.1/IKE	Cryptographic Key Generation (for IKE Peer Authentication)
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_IPSEC_EXT.1	IPsec Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHS_EXT.1	SSH Server Protocol
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1/Functions	Management of Security Functions Behaviour
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMF.1/VPN	Specification of Management Functions (VPN Gateway)
FMT_SMR.2	Restrictions on security roles
FPF_RUL_EXT.1	Rules for Packet Filtering
FPT_FLS.1/SelfTest	Fail Secure (Self-Test Failures)
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

Requirement	Description
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF Testing
FPT_TST_EXT.3	Self-Tests with Defined Methods
FPT_STM_EXT.1	Reliable Time Stamps
FPT_TUD_EXT.1	Trusted Update
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_TAB.1	Default TOE Access Banner
FTP_ITC.1	Inter-TSF Trusted Channel
FTP_ITC.1/VPN	Inter-TSF Trusted Channel (VPN Communications)
FTP_TRP.1/Admin	Trusted Path

5.1 Conventions

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

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

5.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1

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

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

- *[no other actions];*

d) *Specifically defined auditable events listed in* Table 10

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 10.

Table 10 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None
FCS_CKM.1	None	None
FCS_CKM.2	None	None
FCS_CKM.4	None	None
FCS_COP.1/DataEncryption	None	None
FCS_COP.1/SigGen	None	None
FCS_COP.1/Hash	None	None
FCS_COP.1/KeyedHash	None	None
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_IPSEC_EXT.1 (VPN)	Session Establishment with peer	Entire packet contents of packets transmitted/received during session establishment
FCS_RBG_EXT.1	None	None
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded	Origin of the attempt (e.g., IP address)
FIA_PMG_EXT.1	None	None
FIA_UIA_EXT.1	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UAU_EXT.2	All use of identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UAU.7	None	None
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store

Requirement	Auditable Events	Additional Audit Record Contents
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None	None
FMT_MOF.1/Functions	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None
FMT_MTD.1/CoreData	None	None
FMT_MTD.1/CryptoKeys	None	None
FMT_SMF.1	All management activities of TSF data	None
FMT_SMR.2	None	None
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	• Source and destination addresses • Source and destination ports • Transport Layer Protocol
FPT_SKP_EXT.1	None	None
FPT_APW_EXT.1	None	None
FPT_TST_EXT.1	None	None
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None
FTA_SSL.3	The termination of a remote session by the session locking mechanism	None
FTA_SSL.4	The termination of an interactive session	None
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism	None
FTA_TAB.1	None	None
FTP_ITC.1	• Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel functions	Identification of the initiator and target of failed trusted channels establishment attempt

Requirement	Auditable Events	Additional Audit Record Contents
FTP_TRP.1/Admin	• Initiation of the trusted path • Termination of the trusted path. • Failure of the trusted path functions.	None

5.2.1.2 FAU_GEN.2 User Identity Association

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

5.2.1.3 FAU_STG_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: *[oldest audit log is overwritten when a new audit record is generated in a circular fashion]*] when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

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

- ECC schemes using “NIST curves” [P-256, P-384] 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].

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

5.2.2.2 FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication)

FCS_CKM.1.1/IKE[1]

The TSF shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with a specified cryptographic key generation algorithm: **[**

- **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and [no other curves]]**

and [

- **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]**

]

and specified cryptographic key sizes [equivalent to, or greater than, a symmetric key strength of 112 bits].

5.2.2.3 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: [selection:

- 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

5.2.2.4 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-pass] *overwrite consisting of* [a new value of the key];

that meets the following: *No Standard*

[1] Modified in accordance with VPNGW selections for CSfC.

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

FCS_COP.1.1/DataEncryption
The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in **[CBC, GCM], and [No other]** mode and cryptographic key sizes **[256 bits], and [no other cryptographic key sizes]** that meet the following: AES as specified in ISO 18033-3, **[CBC as specified in ISO 10116, GCM as specified in ISO 19772], and [no other standards]**.

5.2.2.6 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 size (modulus) [*3072 bits or greater],*
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits*, 384 bits]

]
that meet the following: [

- *For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS2v1_5; ISO/IEC 9796-2, Digital signature scheme2 or Digital Signature scheme 3,*
- *For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384]; ISO/IEC 14888-3, Section 6.4*

].

5.2.2.7 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-256, SHA-384] ~~and cryptographic key sizes~~ [~~*assignment: cryptographic key sizes*~~] and **message digest sizes [256, 384] bits** that meet the following: *ISO/IEC 10118-3:2004*.

5.2.2.8 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-384] and cryptographic key sizes [384] **and message digest sizes [384] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.2.9 FCS_IPSEC_EXT.1 IPsec Protocol

FCS_IPSEC_EXT.1.1
The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2
The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched and discards it.

FCS_IPSEC_EXT.1.3
The TSF shall implement [tunnel mode].

FCS_IPSEC_EXT.1.4
The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [**AES-CBC-256 (RFC 3602)] and [no other algorithm]** together with a Secure Hash Algorithm (SHA)-based HMAC [HMAC-SHA-384].

FCS_IPSEC_EXT.1.5
The TSF shall implement the protocol: [
- IKEv2 as defined in RFC 5996 and [with no support for NAT traversal], and [RFC 4868 for hash functions]

].

FCS_IPSEC_EXT.1.6
The TSF shall ensure the encrypted payload in the [IKEv2] protocol uses the cryptographic algorithms [AES-CBC-256 (specified in RFC 3602)].

FCS_IPSEC_EXT.1.7
The TSF shall ensure that [selection:
- IKEv2 SA lifetimes can be configured by a Security Administrator based on
  [
    - length of time, where the time values can be configured within [5 *minutes up to 24] hours*

  ]

].

FCS_IPSEC_EXT.1.8
The TSF shall ensure that [
- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on
  [
    - number of bytes;
    - length of time, where the time values can be configured within [5 *minutes up to 8*] hours;

  ]

].

FCS_IPSEC_EXT.1.9[2]
The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least **[*256*]** bits.

FCS_IPSEC_EXT.1.10
The TSF shall generate nonces used in [IKEv2] exchanges of length [
- [at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash].

FCS_IPSEC_EXT.1.11[3]
The TSF shall ensure that IKE protocols implement DH Group(s)
- **19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and [**
- [no other DH groups] according to RFC 5114
  ].

[2] Modified in accordance with VPNGW selections for CSfC.
[3] Modified in accordance with VPNGW selections for CSfC.

FCS_IPSEC_EXT.1.12
The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv2 IKE_SA] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv2 CHILD_SA] connection.

FCS_IPSEC_EXT.1.13
The TSF shall ensure that all IKE protocols perform peer authentication using [ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [no other method].

FCS_IPSEC_EXT.1.14
The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: **Distinguished Name (DN),** *[*SAN: Fully Qualified Domain Name (FQDN)].

5.2.2.10 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-256)].

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[1]* platform-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

5.2.2.11 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, 5647].

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

FCS_SSHS_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than *[262144]* 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: [_aes256-gcm@openssh.com].

FCS_SSHS_EXT.1.5

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

FCS_SSHS_EXT.1.6

The TSF shall ensure that the SSH transport implementation uses [implicit] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7

The TSF shall ensure that [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.

5.2.3 Identification and Authentication (FIA)

5.2.3.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within *[1 to 25]* unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

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

5.2.3.2 FIA_PMG_EXT.1 Password Management

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

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

b) Minimum password length shall be configurable to between [*15*] and [*128*] characters.

5.2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

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

- Display the warning banner in accordance with FTA_TAB.1;

- [*No other actions*]

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

5.2.3.4 FIA_UAU_EXT.1 Password-based Authentication Mechanism

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

5.2.3.5 FIA_UAU.7.1 Protected Authentication Feedback

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

5.2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates**.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [Certificate Revocation List (CRL) as specified in RFC 5759 Section 5].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

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

5.2.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **IPsec and [no other protocols]**, 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.

5.2.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

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

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

5.2.4 Security Management (FMT)

5.2.4.1 FMT_MOF.1/Functions Management of Security Functions Behavior

FMT_MOF.1.1/Functions
The TSF shall restrict the ability to modify the behavior of the function *transmission of audit data to an external IT entity*.

5.2.4.2 FMT_MOF.1/Manual Update Management of Security Functions Behavior

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

5.2.4.3 FMT_MTD.1/CoreData Management of TSF Data

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

5.2.4.4 FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys
The TSF shall restrict the ability to [[*manage*]] the [*cryptographic keys* ***and certificates used for VPN operation***] to [*Security Administrators*].

5.2.4.5 FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using* ***digital signatures and [no other]*** *capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- ***Ability to manage the cryptographic keys;***
- ***Ability to configure the cryptographic functionality;***
- ***Ability to configure the lifetime for IPsec Sas;***
- ***Ability to import X.509v3 certificates to the TOE's trust store;***
    - *[Ability* to modify the behavior of the transmission of audit data to an external IT entity;

- Ability to set the time which is used for time-stamps;
- IPsec Ability to configure the reference identifier for the peer;
- Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors;

].

5.2.4.6 FMT_SMF.1.1/VPN Specification of Management Functions (VPN Gateway)

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

- *Definition of packet filtering rules;*
- *Association of packet filtering rules to network interfaces;*

Ordering of packet filtering rules by priority;

- [No other capabilities]].

5.2.4.7 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator.*

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

FMT_SMR.2.3
The TSF shall ensure that the conditions

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

are satisfied.

5.2.5 Packet Filtering (FPF)

5.2.5.1 FPF_RUL_EXT.1.1 Rules for Packet Filtering

FPF_RUL_EXT.1.1

The TSF shall perform Packet Filtering on network packets processed by the TOE.

FPF_RUL_EXT.1.2

The TSF shall allow the definition of Packet Filtering rules using the following network protocols and protocol fields:

- Ipv4 (RFC 791)
  - Source address
  - Destination Address
  - Protocol
- Ipv6 (RFC 2460)
  - Source address
  - Destination Address
  - Next Header (Protocol)
- TCP (RFC 793)
  - Source Port

    - Destination Port
- UDP (RFC 768)
    - Source Port
    - Destination Port

FPF_RUL_EXT.1.3

The TSF shall allow the following operations to be associated with Packet Filtering rules: permit and drop with the capability to log the operation.

FPF_RUL_EXT.1.4

The TSF shall allow the Packet Filtering rules to be assigned to each distinct network interface.

FPF_RUL_EXT.1.5

The TSF shall process the applicable Packet Filtering rules (as determined in accordance with FPF_RUL_EXT.1.4) in the following order: Administrator-defined.

FPF_RUL_EXT.1.6

The TSF shall drop traffic if a matching rule is not identified.

5.2.6 Protection of the TSF (FPT)

5.2.6.1 FPT_APW_EXT.1 Protection of Administrator Passwords

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

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

5.2.6.2 FPT_FLS.1/SelfTest Fail Secure (Self-Test Failures)

FPT_FLS.1.1/SelfTest
The TSF shall **shut down** when the following types of failures occur: [*failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests*].

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

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

5.2.6.4 FPT_STM_EXT.1 Reliable Time Stamps

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

FPT_STM_EXT.1.2
The TSF shall [allow the Security Administrator to set the time]

5.2.6.5 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [during initial startup (on power on)] to demonstrate the correct operation of the TSF: [*noise source health tests*, *the self-tests defined in Table 11*].

Table 11 – Cryptographic Self-Tests

Crypto Implementation	Self-Tests
U-Boot and Xilinx UltraScale+ MPSOC	Firmware integrity test using RSA 4096 with SHA-384
BoringSSL	AES KAT (encryption and decryption) ECDSA KAT (signature generation/signature verification) ECDSA Pairwise Consistency Test HMAC-SHA-512 KAT RSA KAT (signature verification, and encryption/decryption) SHA-512 KAT
Secure Element Chip	A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.
Linux Kernel	AES KAT (encryption and decryption) AES-CBC KAT HMAC-SHA384 KAT SHA-384 KAT

5.2.6.6 FPT_TST_EXT.3 TSF Self-Test with Defined Methods

FPT_TST_EXT.3.1

The TSF shall run a suite of the following self-tests [*[when loaded for execution]*] to demonstrate the correct operation of the TSF: [*integrity verification of stored executable code*].

FPT_TST_EXT.3.2
The TSF shall execute the self-testing through [*a TSF-provided cryptographic service specified in FCS_COP.1/SigGen*].

5.2.6.7 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 mechanism and** [**no other mechanisms**] prior to installing those updates.

5.2.7 TOE Access (FTA)

5.2.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

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

5.2.7.2 FTA_SSL.3 TSF-initiated Termination

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

5.2.7.3 FTA_SSL.4 User-initiated Termination

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

5.2.7.4 FTA_TAB.1 Default TOE Access Banner

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

5.2.8 Trusted Path/Channels (FTP)

5.2.8.1 FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1
The TSF shall **be capable of using [IPsec] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [VPN peers]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data**.

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

FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for [remote VPN gateways/peers*]*.

5.2.8.2 FTP_ITC.1/VPN Inter-TSF Trusted Channel (VPN Communications)

FTP_ITC.1.1/VPN

The TSF shall **be capable of using IPsec to** provide a communication channel between itself and **authorized IT entities supporting VPN communications** 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/VPN

The TSF shall permit [*the authorized IT entities*] to initiate communication via the trusted channel.

FTP_ITC.1.3/VPN

The TSF shall initiate communication via the trusted channel for [remote VPN gateways/peers].

5.2.8.3 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*.

5.3 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.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the PP and any relevant Modules, which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 12.

Table 12 – 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

5.5 Assurance Measures

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

Table 13 – TOE Security Assurance Measures

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

6 TOE Summary Specification

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

Table 14 – TOE Summary Specification

Requirement	TSS Description
FAU_GEN.1 FAU_GEN.2 FAU_STG_EXT.1	The TOE generates audit records containing the subject identity that caused the event (if applicable), timestamp, description, and outcome. The auditable events are described in Table 10 in FAU_GEN.1 above. The TOE is standalone and stores audit data locally, though remote logging is also supported via device telemetry that is transmitted to the trusted network. Audit data is recorded locally on the TOE under /var/log/messages, and these data are transmitted over UDP packets in real-time for ingestion by a logging service running on a trusted node, which is protected via IPSec. ~64MB total of data can be stored locally, though logs are additionally transmitted to long term storage via UDP telemetry on the trusted of the network via IPSec. When the audit data store is full, the log file discards the oldest messages to store the new. The records are protected against unauthorized access and modification on the local TOE by root administrative SSH access, which is additionally gated by trusted network access. Local logs are stored in /var/log/messages which has a maximum size of ~64M. Once the /var/log/messages file reaches 1M in size, the log rotator shifts the old logs to /var/log/messages.1, and creates a fresh /var/log/messages file. This process continues, with messages.1 moving to messages.2 to messages.3 (etc.) on each rotation until the default maximum of messages.5. Once there are 5 files, the next rotate deletes the oldest messages log file. When the SSH keys are modified, the changes will trigger logging by the audit daemon, which captures administrator or user-initiated changes to the file system, reporting to /var/log/messages. SSH keys are identified by associating to a specific user account. For any key rotation of the private identity keys, the changes are logged as kernel messages to /var/log/messages. Private identity keys are identified by the certificate to which they belong.
FCS_CKM.1 FCS_CKM.1/IKE	The TOE's cryptographic key material for IKEv2 and SSH is generated using the Secure Element Chip. It is capable of generating P-256, and P-384 bit keys using approved FIPS 186-4 ECDSA key generation in accordance with Appendix B.4 with no omissions or implementation-specific extensions.
FCS_CKM.2	The TOE supports SP 800-56Arev3 key establishment using ECDH and DH in support of IKEv2 session establishment (FCS_IPSEC_EXT.1). The TOE supports Diffie-Hellman Group 19 and Group 20. The TOE supports ecdh-nistp256 and ecdh-nistp384 for SSH key exchange (FCS_SSHS_EXT.1).
FCS_CKM.4	Refer to section 6.2 for a listing of all relevant keys utilized by the TOE including their origin and storage location, and applicable key destruction situations.

Requirement	TSS Description
	During normal operation of the TOE, there should not be a situation where the keys are not successfully destroyed. Any keys present in volatile memory are ephemeral, and would have no use if the TOE were to experience a power failure that would prevent the proper destruction of the keys. New ephemeral keys would be created during proper runtime of the TOE after a successful restart.
FCS_COP.1/DataEncryption	The TOE supports AES in CBC and GCM modes with 256-bit keys in support of IPsec and SSH, respectively.
FCS_COP.1/Hash	SHA-256 and SHA-384 hashing algorithms are associated with ECDSA P-256 and ECDSA P-384 signatures respectively. The TOE implements SHA-384 with HMAC-SHA-384 for IKE/IPsec and SSH.
FCS_COP.1/KeyedHash	The TOE implements HMAC-SHA-384 in support of IPsec which has key length 256, block size 128, and output MAC length is 48 bytes The TOE implements HMAC-SHA-384 in support of SSH, which has key length 256, block size of 128, and output length 64 bytes or 32 bytes
FCS_COP.1/SigGen	Cryptographic signing is handled by ECDSA P-256 and ECDSA P-384 in the Secure Element chip on data that has been hashed with SHA-256 and SHA-384 in support of IPsec and SSH. RSA 4096 signatures with SHA-384 are used to verify image integrity on boot and for software upgrades.
FCS_IPSEC_EXT.1 FPF_RUL_EXT.1 FIA_X509_EXT.1/Rev FIA_X509_EXT.2 FIA_X509_EXT.3	The TOE implements IKEv2 with X.509 EC certificates to establish IPsec (tunnel mode only) connections to remote VPN endpoints including endpoints with audit log receiver services to protect the transmission of audit records. The TOE implements AES-256-CBC and HMAC-SHA-384 algorithms for both IKEv2 and ESP. As AES-256-CBC is the TOE's strongest strength of encryption in terms of key size , the negotiated IKEv2 SA cipher is sufficient to protect the IKEv2 CHILD_SA connection and no weaker key strengths than 256 will be accepted. The TOE supports Diffie-Hellman Group 19 (with a key size of 256 bits) and Group 20 (with a key size of 384 bits) for IKEv2 key exchange. The nonces are generated using the Secure Element Chip's DRBG with the appropriate lengths in accordance with the negotiated DH group. The 'x' in g^x mod p is generated using the DRBG in accordance with the negotiated DH group. Lifetime is configurable by setting the TOE's ikelifetime parameter for the IKEv2 SA using an integer value with 'm' and 'h' for minutes and hours. The supported range is between 5 minutes and 24 hours. Lifetime for the child SA is configurable by setting the lifetime parameter using an integer value with 'm' and 'h' for minutes and hours. The supported range is between 5 minutes and 8 hours. During the IKEv2 negotiation, an FQDN is used as the matching constraint in the configuration files. This FQDN will appear as a Subject Alternative Name (SAN) in the X.509v3 certificate. SAN is prioritized over CN. The TOE

Requirement	TSS Description
	supports generation of certificate signing requests using the following fields: ● C = US, O = [Organization Name], CN = [IP Address] In addition to ensuring that the subject identifier of the peer matches the expected identifier in the TOE's configuration, the peer certificate presented during the IKEv2 exchange (or the TOE's own certificate during certificate installation) will be validated according to the following rules: ● The certificate chain is validated using the public key of the issuer's CA certificate, in a recursive process terminating at the root of trust. If a complete certificate chain is not presented, the connection fails. ● All certificates in the chain must not be expired. If any certificate is expired, the connection fails. ● The peer VPN certificate and all intermediate CAs in the chain must not be revoked. During the IKEv2 negotiation, the CRL file identified by each certificate is checked against a CRL cache which is refreshed regularly through the update process. If the CRL validity period is expired, or a CRL cache is not present, or if any of the intermediate CAs or endpoint certificates are revoked, the connection fails. When loading a CA certificate, the basicConstraints flag must be present and set to TRUE, otherwise, the operation will fail. The TOE supports only one certificate for itself. The TOE implements a security policy database (SPD). The iptables rules and the TOE's tunneling configuration form the SPD in terms of the BYPASS (e.g., no encryption), DISCARD (e.g., drop the packet), and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301. The processing of entries in the SPD works as follows. The SPD may contain multiple entries, each with a different access list. The entries are parsed in a sequence starting with the iptables rules where it determines if a match to the ACL is found for a given packet, where the action may be 'drop' or 'allow'. If the packet is tagged as an IPsec packet, a new SA is established (if no matching SA exists and IKEv2 is able to authenticate the peer successfully) or the packet is processed according to an existing SA, and thus the packet would be 'protected'. Packets not tagged as IPsec packets, but otherwise allowed by an iptables rule would bypass the tunnel in plaintext. Iptables rules are configurable by an authorized administrator. Packets which match a filter rule with logging are audited in the TOE audit trail. When network packets are received at any TOE interface, the TOE verifies whether the network traffic is allowed or not, and performs one of the following actions: Allow or Drop, with or without logging. By implementing rules that define the permitted flow of traffic between interfaces of the TOE, the administrator may control when and how

Requirement	TSS Description
	packets are handled by the TOE. Packets may be filtered based on any of the following: • IPv4 or IPv6 address of source • IPv4 or IPv6 address of destination • Transport layer protocol or IPv6 next header. • Service used (TCP or UDP, both source and destination) • Network interface on which the packet is received These rues support the following protocols: RFC 791 for IPv4, RFC 2460 for IPv6, RFC 793 for TCP, and RFC 768 for UDP. Compliance with these standards is verified via interoperability and regression testing. Packets will be dropped unless a specific rule has been set up to allow the packet to pass (where the attributes of the packet match the attributes in the rule and the action associated with the rule is to pass traffic). This is the default action that occurs on an interface if no packet filtering rule is found. If a packet arrives that does not meet any rule, it is dropped. Rules are enforced on a first match basis from the top down. As soon as a match is found the action associated with the rule is applied. Packet Filtering rules are operational as soon as interfaces are operational following startup of the TOE. There is no state during initialization/ startup that the access lists are not enforced on an interface. The initialization process first initializes the operating system, and then the networking daemons including the packet filtering enforcement, prior to any daemons or user applications that potentially send network traffic. No incoming network traffic can be received before the packet filtering functionality is operational. In the event that a network interface is overwhelmed by too much incoming network traffic, the TOE will silently drop packets which it cannot process, without logging them to the audit trail. Under no circumstances will the TOE allow a packet to pass if it does not satisfy an existing “Permit” rule or belong to an established session.
FCS_RBG_EXT.1	The TOE uses the CTR_DRBG with AES-256 for all key generation. The DRBG is seeded with minimum of 256 bits which is seeded with entropy from the Secure Element Chip (platform-based noise source).
FCS_SSHS_EXT.1	The TOE support SSHv2 for remote administration with the following characteristics: • Public key authentication is verified by checking that the presented key matches one of the options stored in the authorized_keys file and supports ecdsa-sha2-nistp384 as the public key algorithm as well as the host key algorithm. Public keys are prioritized before keyboard-authentication with a password. • Password-based authentication is verified using the Linux PAM framework using a stored username and password combination. • Packets larger than 262144 bytes will be dropped. • AES-256-gcm is the only supported cipher and authentication algorithm • ecdh-sha2-nistp256 and ecdh-sha2-nistp384 are the only key exchange methods supported.

Requirement	TSS Description
	• Rekeying is implemented according to the default RekeyLimit of 512M (megabytes) and 2700 (seconds).
FIA_AFL.1 FIA_PMG_EXT.1 FIA_UIA_EXT.1 FIA_UAU_EXT.2 FIA_UAU.7	The TOE supports administrative login to a CLI through a local UART serial console and SSH. No non-administrative users are supported and thus have no access to functionality. Prior to authentication of an administrative user via password (local) or via password/SSH public key (remote), a login banner is displayed. For password authentication, the administrator supplies a valid username and password. If the username matches a record in the /etc/shadow file, and the password digest matches, the authentication succeeds. If either the username or password do not match, the authentication fails and the failed login counter is incremented. Once the counter reaches the administrator-configured limit, the account is locked and cannot be used to login again until an administrator-configurable amount of time has elapsed. Lockouts are not enforced at the local console, nor are failed attempts tracked. For public key authentication, the client signs a nonce using the SSH private key which is verified using the public key stored in the "authorized_keys" file. If the data cannot be verified, the authentication fails. Note: the failed login counter is not incremented for local console access attempts or SSH public key authentication attempts. No password feedback is displayed for CLI login sessions as credentials are being entered. Passwords must be constructed using a configurable minimum length between 15 and 128 characters. The TOE supports the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")".
FMT_MTD.1/CoreData FMT_MTD.1/CryptoKeys FMT_MOF.1/Functions FMT_SMF.1 FMT_SMR.2	The TOE supports remote administration via SSH CLI and local administration through a serial UART CLI. The TOE supports only one role, that of "Security Administrator". All security management functionality is restricted to the Security Administrator role. All management functions may be performed either via SSH or local console. The following management functionality is supported: • Local and remote administration via CLI • Local and remote login banner configuration • Local and remote session inactivity timeout configuration • Manual software upgrade • Authentication failure lockout configuration • Cryptographic functionality and key management (Generation and deletion, via the appropriate interfaces in the underlying operating system) • IPsec lifetime configuration • IPsec peer identifier configuration • Management of the IPsec tunnel used to forward audit data • X.509 certificate management including import and designation of certificates as trust anchors • Time and date configuration

Requirement	TSS Description
FPT_APW_EXT.1	TOE passwords are obfuscated with a salted hash, and stored in the underlying file system. The TOE does not provide any interface which would allow a user or administrator to directly view the private portion of any key, or any password in plaintext.
FPT_SKP_EXT.1	Please refer to section 6.2 for detail on how any symmetric keys and private keys are stored. EC identity keys are protected by the Secure Element Chip and are not accessible through any interface. Keys stored in non-volatile storage are protected with filesystem permissions and are not accessible to users.
FPT_STM_EXT.1	The TOE contains a real-time clock and supports manual time configuration by the Security Administrator to provide reliable timestamps for: ● audit logging ● X.509 certificate validity ● SSH and IPsec rekey operations ● session timeouts
FPT_TST_EXT.1 FPT_FLS.1/SelfTest	The following self-tests are performed at power-on: ● Firmware Integrity Test (RSA 4096 bits with SHA-384) ● BoringSSL ○ AES KAT (encryption and decryption) ○ ECDSA KAT (signature generation/signature verification) ○ ECDSA Pairwise Consistency Test ○ HMAC-SHA-256 KAT ○ HMAC-SHA-512 KAT ○ RSA KAT (signature generation/signature verification and encryption/decryption) ○ SHA-256 KAT ○ SHA-512 KAT ● Secure Element Chip self-test ○ A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output. Kernel Cryptography ○ AES KAT (encryption and decryption) ○ AES-CBC KAT ○ HMAC-SHA384 KAT ○ SHA-384 KAT Cryptographic known answer testing is performed by using a known seed to derive an expected key, or using a known key to encrypt/decrypt a known value. If the result of the process is anything other than as expected, the test fails. Firmware integrity is verified by performing a cryptographic hash of the executable code and comparing the result to a known hash value; if the result is a difference, the firmware is known to have become corrupt. If any of these test fail, the TOE halts operation and reboots into a networking-disabled “safe mode”. If any of the Known Answer Tests (KAT) were to fail, the TOE would fail to perform a function as soon as the TOE

Requirement	TSS Description
	initialized the cryptographic library. This would result in the TOE being unreachable over SSH or IPsec. If the pairwise consistency test failed, the shutdown of TOE functionality would occur during an ECDSA negotiation, such as when trying to establish a new SSH connection (whether a successful one has already been performed or not) or when performing an IKEv2 SA or child SA negotiation.. Because the TOE will not boot into a normal executing mode if any self-tests fail, these tests are sufficient to demonstrate that the TSF is operating correctly.
FPT_TUD_EXT.1 FMT_MOF.1/ManualUpdate	The version of firmware running on the TOE can be checked by running the 'version_info' command. The TOE supports only one firmware image to be installed on the device at any given time and therefore does not support delayed activation. Updates are performed manually by placing a valid update package on the TFTP server and triggering a restart of the TOE. Upon restarting, the new package will be ingested by the TOE where it will perform an RSA-4096 with SHA-384 signature validation of the image. If the signature is successfully verified, the TOE will boot up into its operational state with the new version. If the signature cannot be validated, or if no signature is present, the TOE will not load the new firmware and will boot up using the existing firmware on the device. Administrators should contact the vendor support for assistance. Update candidates are obtained from the vendor's content delivery network, and must be manually carried to an administrator-controlled TFTP server for transfer to the TOE. The TFTP service itself is not in-scope of the evaluation. Configuration of the TFTP destination is done only by a properly authorized and authenticated administrator. The TOE initiates all TFTP connections to the proper destination, and does not allow or accept TFTP connections from outside peers. The TFTP service is hard-coded to restrict its functionality such that only update candidates are obtained from the server, and update candidates are subjected to the signature validation mechanisms described above to ensure that only properly signed, valid, updates are applied by the TOE.
FTA_SSL_EXT.1 FTA_SSL.3	The TOE supports session termination of the UART serial console after a period of inactivity. This timeout setting is configured by CLI command. The TOE supports session termination of the SSH after a period of inactivity. This timeout setting is configured by CLI command.
FTA_SSL.4	SSH and serial console CLI sessions may be terminated using the "logout" or "exit" command.
FTA_TAB.1	The TOE supports local administration via UART serial console, and remote administration via SSH. Both methods of access support displaying a configurable advisory and consent notice to administrators prior to authentication.
FTP_ITC.1	The TOE supports a trusted channel to the following authorized IT entities:

Requirement	TSS Description
	• VPN peers via IPsec (peer-to-peer) using the protocols defined in [ST] section 5.2.2.9 • Audit log receiver via IPsec (client) using the protocols defined in [ST] section 5.2.2.9
FTP_TRP.1/Admin	The TOE supports a trusted path from administrators via SSHv2 using password-based and public-key based authentication and the SSH protocol defined by the selections in [ST] section 5.2.2.11

6.1 CAVP Algorithm Certificate Details

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

Table 15 – CAVP Algorithm Certificate References

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #	TSF Supported
FCS_CKM.1	ECC schemes using "NIST curves" [selection: P-256, P-384] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	SpaceX Cryptographic Module (BoringSSL)	ECDSA KeyGen	A3452	Public/Private key creation for X.509v3 Certificate CSRs. Peer authentication for IPsec Peer authentication for SSH
	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]	N/A	Vendor Affirmed	N/A	EC DH Key exchange for SSH EC DH Key exchange for IPsec
FCS_CKM.2	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"	SpaceX Cryptographic Module (BoringSSL)	ECDSA Key Establishment	A3452	Ephemeral Key creation for IPsec Ephemeral Key creation for SSH
	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]	N/A	Vendor Affirmed	N/A	EC DH Key exchange for SSH
FCS_COP.1/ DataEncryption	AES used in [CBC, GCM] mode and cryptographic key sizes [256 bits]	SpaceX Cryptographic Module (BoringSSL)	AES-CBC-256 AES-GCM-256	A3452	Bulk Encryption for IPsec Phase 1 SA's Bulk Encryption for SSH
		SpaceX Linux Kernel cryptographic Module	AES-CBC-256	A3121	Bulk Encryption for IPsec Phase 2 SA's.
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	SpaceX Boot Loader cryptographic Module (First-Stage Boot Loader)	RSA SigVer 4096	A3120	RSA digital signature verification for trusted update.

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #	TSF Supported
	signature scheme 2 or Digital Signature scheme 3				
	For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384]; ISO/IEC 14888-3, Section 6.4	SpaceX Cryptographic Module (BoringSSL) NXP JCOP4 EC Component For N7121 (Secure Element Chip)	ECDSA SigGen, sigVer over P-256 or P-384	A3452 C1429	ECDSA Signature operations for IKEv2 SA's ECDSA signature operations for SSH public key authentication.
FCS_COP.1/ Hash	[SHA-256, SHA-384] and message digest sizes [256, 384] bits	SpaceX Cryptographic Module (BoringSSL) SpaceX Uboot cryptographic Module SpaceX Cryptographic Module (BoringSSL)	SHA2-384 SHA2-256 SHA2-384	A3452 A3122 A3452	IPsec Phase 1 SA hashing Trusted Update hash comparison, firmware integrity verification ECDH-SHA2 key exchange in SSH
FCS_COP.1/ KeyedHash	[HMAC-SHA-384,] and cryptographic key sizes [384] and message digest sizes [384] bits	SpaceX Cryptographic Module (BoringSSL) SpaceX Linux Kernel cryptographic Module	HMAC-SHA2-384	A3452 A3121	Message authentication codes for IPsec SA's
FCS_RBG_EXT.1	CTR_DRBG (AES-256)	NXP JCOP4 DRBG Component For N7121 (Secure Element Chip)	CTR_DRBG with AES-256	C886	DRBG for all TOE functions.

6.2 Cryptographic Key Destruction

The table below describes the key zeroization provided by the TOE and as referenced in FCS_CKM.4. All password destruction which is described as "automatic" occurs at the indicated condition and is performed by underlying operating system functions.

Table 16 – Cryptographic Keys

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
skeyid	IKE intermittent value used to create skeyid_d.	RAM	Automatically after used to generate key materials. Overwritten with zeroes.
skeyid_d	IKE intermittent value used to derive keying data for IPsec.	RAM	Automatically after IKE session terminated. Overwritten with zeroes.

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
DH/ECDH Shared secret	Shared secret calculated as a result of Diffie-Hellman key exchange	RAM	Automatically after IKE/SSH session terminated. Overwritten with zeroes.
DH/ECDH Private exponent	private exponent used as part of the Diffie-Hellman key exchange.	RAM	Automatically after IKE/SSH session terminated. Overwritten with zeroes.
IKE session encryption key	AES key used for encrypting Phase 1 SAs.	RAM	Automatically after IKE session terminated. Overwritten with zeroes.
IKE session authentication key	HMAC key used for authenticating Phase 1 SAs.	RAM	Automatically after IKE session terminated. Overwritten with zeroes.
IPsec session encryption key	AES key used for encrypting Phase 2 SAs.	RAM	Automatically after IPsec session terminated. Overwritten with zeroes.
IPsec session authentication key	HMAC key used for authenticating Phase 2 SAs.	RAM	Automatically after IPsec session terminated. Overwritten with zeroes.
ECDSA or RSA private key	Private signing key used for IKE and SSH authentication	Secure Element Chip	When a new keypair is generated, the TOE performs a single-pass overwrite consisting of a new value of the key.
SSH session encryption key	AES key used for encrypting Phase 2 SAs.	RAM	Automatically after SSH session terminated. Overwritten with zeroes.
SSH session authentication key	HMAC key used for authenticating Phase 2 SAs.	RAM	Automatically after SSH session terminated. Overwritten with zeroes.
Administrator password	Password used to authenticate via SSH or UART	Plaintext password: RAM Password Hash: Flash	Password hash is persistent on disk Plaintext password is overwritten with zeros by

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
			PAM after being obtained by SSH or UART Plaintext password is overwritten by login after user session is established
Entropy input	Seed values for the DRBG	Secure Element Chip	Automatically after DRBG reseed. Overwritten with zeroes.

7 Acronym Table

Acronyms should be included as an Appendix in each document.

Table 17 - Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CC	Common Criteria
CLI	Command Line Interface
CRL	Certificate Revocation List
CTR	Counter
DRBG	Deterministic Random Bit Generator
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
GCM	Galois Counter Mode
HMAC	Hash Based Message Authentication Code
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	Internet Protocol Security
NDcPP	Network Device Collaborative Protection Profile
KAT	Known Answer Test
NIAP	Nation Information Assurance Partnership
PP	Protection Profile
OSP	Organizational Security Policy
RAM	Random Access Memory
RSA	Rivest, Shamir & Adleman
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SSH	Secure Shell
SoC	System on Chip
ST	Security Target
TFTP	Trivial File Transfer Protocol
TOE	Target of Evaluation
TLS	Transport Layer Security
TSS	TOE Summary Specification
UART	Universal Asynchronous Receiver/Transmitter
VPN	Virtual Private Network