HYCU for Enterprise Clouds Security Target

Document Version: 0.2.9

intertek
acumen
security

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History

Version	Date	Changes
Version 0.1	April 18th 2022	Initial Draft
Version 0.1.2	July 14th 2022	Updated by HYCU
Version 0.1.3	July 15, 2022	Updates to TSS
Version 0.1.4	Aug 9th, 2022	Updated by HYCU based on Acumen review
Version 0.1.5	Aug 25th 2022	Minor rewording in section 1.3.2, added TD0638, removed previous markings, updates to SFR and TSS sections.
Version 0.1.6	September 20, 2022	Updated by HYCU based on Acumen review: updated TSS sections and answered questions to comments.
Version 0.1.7	September 21, 2022	Updated following QA comments, added TD0639 and TD0670.
Version 0.1.8	September 28, 2022	Updated by HYCU based on Acumen review, updated TSS and answered questions to comments.
Version 0.1.9	September 30, 2022	Minor updates to TSS.
Version 0.2.0	October 26th, 2022	TSS updates, added CAVP certificates.
Version 0.2.1	April 20th, 2023	Updated Cipher List for TLS Claims
Version 0.2.2	July 31st 2023	Addressed ECR comments for Check-in
Version 0.2.3	August 3, 2023	Address round 2 of ECR comments for Check-in
Version 0.2.4	September 20, 2023	Updated TSS activities
Version 0.2.5	September 26, 2023	Updated with QA feedback
Version 0.2.6	September 28, 2023	Updated TSS information
Version 0.2.7	October 25, 2023	Addressed ECR Comments
Version 0.2.8	December 20, 2023	Addressed ECR Comments
Version 0.2.9	January 10, 2024	Addressed 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 are met by the TOE.

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	HYCU for Enterprise Clouds Security Target
ST Version	0.2.9
ST Date	January 10, 2024
ST Author	Acumen Security, LLC.
TOE Identifier	HYCU for Enterprise Clouds
TOE Version	4.5.1
TOE Developer	HYCU Inc.
Key Words	Cloud, Enterprise

1.2 TOE Overview

The TOE is the HYCU, Inc. HYCU for Enterprise Clouds. HYCU for Enterprise Clouds provides application-consistent and virtualization-native data protection, data migration and disaster recovery. HYCU for Enterprise Clouds allows administrators to protect and manage clusters of a virtualized infrastructure with one integrated interface.

HYCU for Enterprise Clouds is a software-based TOE that is installed as a virtual machine. The deployed virtual machine is accessed via a web GUI.

1.3 TOE Description

TOE is HYCU for Enterprise Clouds virtual appliance and management access, LDAP/S, SMTP and DNS. The NTP, storage, and hypervisor are not included in TOE. For a full list see Sections 1.5 and 1.6. The following diagram shows the environment and the evaluated TOE.

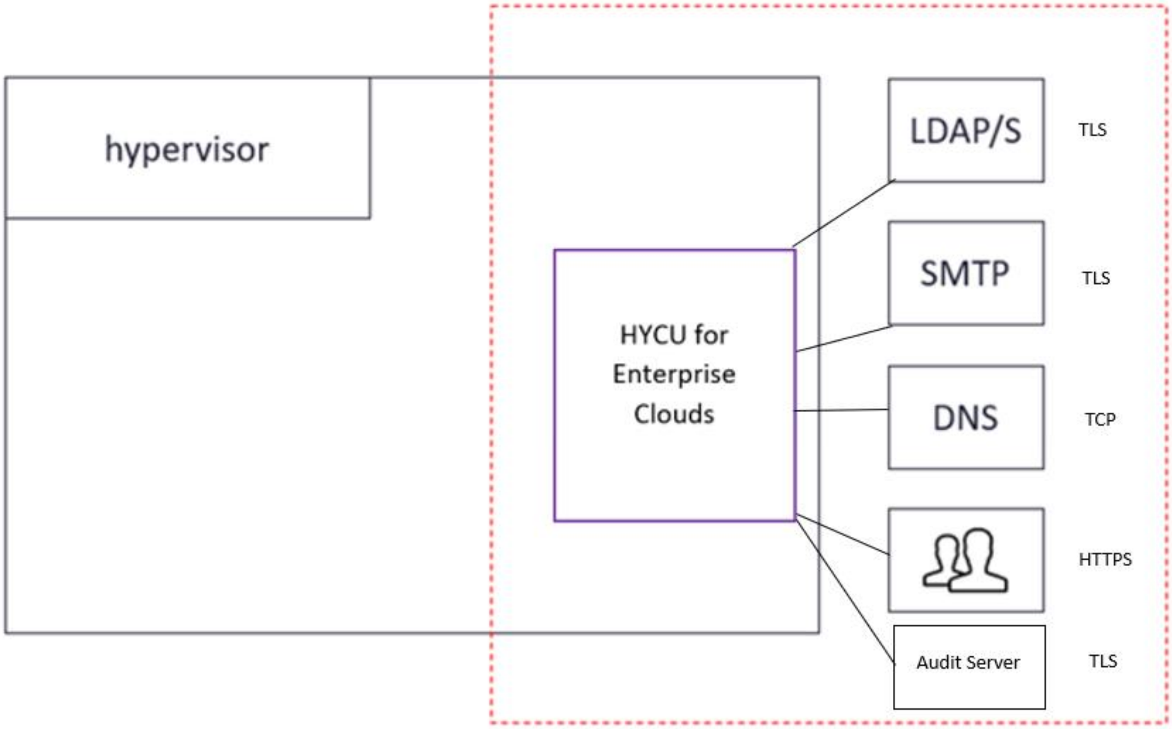

Figure 1 – Representative TOE Deployment

1.3.1 Physical Boundaries

The physical boundaries of the TOE are HYCU for Enterprise Clouds VM running on hypervisor and TOE hardware platform listed in section 1.5. The red dotted line is the evaluated configuration consisting of the HYCU for Enterprise Clouds (TOE) as well as all connections the TOE makes externally (LDAP/S, SMTP/S, Audit Server). The HYCU for Enterprise Clouds is the only VM inside of the VMware ESXi 7 hypervisor.

The TOE runs on an Intel Xeon Silver 4208 (Cascade Lake) processor leveraging VMware ESXi 7.0.3, 19193900 as the hypervisor. The operating system of the TOE is Rocky Linux 8.6 with Java – OpenJDK 1.8.0.332 as the software version.

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.

1.3.2.1 Security Audit

The HYCU for Enterprise Clouds provides extensive auditing capabilities. The TOE generates a comprehensive set of audit logs that identify specific TOE operations. For each event, the TOE records the date and time of each event, the type of event, the subject identity, and the outcome of the event. Auditable events include:

- failure on invoking cryptographic functionality such as establishment, termination and failure of cryptographic session establishments and connections
- modifications to the group of users that are part of the Authorized Administrator roles
- all use of the user identification mechanism
- any use of the authentication mechanism
- administrator lockout due to excessive authentication failures
- any change in the configuration of the TOE
- changes to time
- initiation of TOE update
- indication of completion of TSF self-test
- maximum sessions being exceeded
- termination of a remote or local session
- attempts to unlock a termination session
- initiation and termination of a trusted channel
- failure of the trusted channel functions
- initiation and termination of a trusted path
- failure of the trusted channel path

The TOE is configured to transmit its audit messages to an external audit server. Communication with the audit server is protected using TLS and the TOE can determine when communication with the audit server fails. If that should occur, the TOE will store all audit records locally and when the connection to the remote audit server is restored, all stored audit records will be transmitted to the remote audit server.

The audit logs can be viewed on the TOE. The records include the date/time the event occurred, the event/type of event, the user associated with the event, and additional information of the event and its success and/or failure. The TOE does not have an interface to modify audit records.

1.3.2.2 Cryptographic Support

The TOE utilizes TLS (via HTTPS, SMTP/S and LDAPS) to securely communicate, both with external services (audit server, authentication server, mail server) and external clients (HTTPS for GUI administration). Both RSA and ECDSA keys are supported. Cryptographic support is restricted to the approved set of algorithms using a combination of system-wide policies and application-specific configuration. Random bit generation is served by underlying OS facilities (/dev/random).

1.3.2.3 Identification and Authentication

The TOE allows the Administrator to securely login to the management interface using a username and password. Usernames and passwords can be managed within the TOE or delegated to an external authentication server (AD/LDAPS). A lockout period protects against repeated authentication failures. The TOE can be configured with a custom login banner.

The private key and certificate for the TLS server can be imported or generated on the TOE. The TOE can issue a certificate signing request to be signed by an external certificate authority and then imported for use by the TLS server.

Trusted roots can be imported to establish trust with external servers. The TOE validates certificates of external servers – invalid or untrusted certificates result in rejected communication attempt. Certificate Revocation Lists can be used to manage revocation.

1.3.2.4 Security Management

The TOE is managed remotely via a web user interface. Some functionality requires local console access. Roles and groups (tenants) can be defined and the roles can be assigned to users. TOE management is scoped within a built-in "Infrastructure group". The TOE restricts configuration of security-related functions to the Administrator role of the Infrastructure group.

The Administrator is able to perform the following security-related functions:

- start and stop services
- update the TOE
- modify the behavior of the transmission of audit data to an external IT entity
- manage the cryptographic keys
- configure the cryptographic functionality
- set the time which is used for time-stamps
- manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
- import X.509v3 certificates to the TOE's trust store
- configure the session inactivity time before session termination or locking
- ability to configure the authentication failure parameters for FIA_AFL.1
- ability to configure access banner
- ability to update the TOE, and to verify the updates using hash comparison capability prior to installing those updates
- ability to administer the TOE locally and remotely

1.3.2.5 Protection of the TSF

Passwords of TOE-managed users are stored in a non-reversible encoding in the internal database. Private keys and passwords for external services are stored in an encrypted form within the TOE database. Password input is obscured by default (password reveal is optional).

The administrator can set the local TOE time using the console.

The TOE performs power-on self-tests to verify the integrity of the primary application server and supporting components. Self-tests can be performed on-demand.

The TOE has an update mechanism. Before performing updates, the administrator should manually validate the update image using the published hash available via HTTPS.

1.3.2.6 TOE Access

Idle sessions are terminated by the TOE after a configurable period of inactivity. In the web user interface, a short time before the inactivity period expires, a dialog is shown to notify of an impending session termination. The TOE lets the user sign out of the session on demand using a dedicated sign-out button (for web user interface) or the user can terminate the current shell (for the console).

The TOE can be configured with a custom login banner, for both the web user interface and the console.

1.3.2.7 Trusted Path/Channels

The TOE uses TLS to securely communicate with the following authorized IT entities:

- authentication server (Active Directory via LDAP/S)
- mail server (via SMTP/S)
- audit server (via HTTPS webhooks)

Administrator access to the web user interface is protected using TLS (via HTTPS).

1.4 TOE Documentation

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

- HYCU for Enterprise Clouds Security Target v0.2.9 January 2024 (this document)
- HYCU for Enterprise Clouds Administrative Guide December 2023
- HYCU Data Protection for Enterprise Clouds User Guide, Version 4.5.1, July 2022

1.5 TOE Environment

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

Table 2 – Required Environmental Components

Component	Required	Purpose/Description
Lenovo ThinkSystem SR630, Xeon Silver 4208	Yes	TOE hardware platform
VMware ESXi 7	Yes	Hypervisor
LDAP/S	Yes	Remote authentication
SMTP	Yes	Notifications
Administrator Workstation	Yes	Management of the TOE
DNS server	Yes	Name resolution
Audit Server	Yes	Audit Log Transfer

1.6 Product Functionality not Included in the Scope of the Evaluation

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

- Linux and Windows based targets (NFS/CIFS)
- Cloud-based targets (Google, Amazon, Azure)
- iSCSI targets
- File-level recovery
- Reporting
- Nutanix File Server Backup
- VMware Virtual Machine Backup and Physical Machine Backup
- Virtual Machine Backup for Nutanix (AHV and ESXi)
- Application Awareness and Backup (Microsoft Active Directory, Exchange, SQL Server, Oracle Database)

- SSH
- Mutually authenticated TLS
- Encrypted backups
- S3 Compatible Targets
- NTP time synchronization
- Web GUI certificate authentication

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:

- Collaborative Protection Profile for Network Devices, Version 2.2e, 27 March 2020 [CPP_ND_V2.2E]

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), performing only the operations defined there.

2.3.1 Technical Decisions

All NIAP TDs issued to date and applicable to NDcPP v2.2e 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)	Y	
TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	N	NTP is not included in the TOE.
TD0536: NIT Technical Decision for Update Verification Inconsistency	Y	
TD0537: NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Y	
TD0546: NIT Technical Decision for DTLS – clarification of Application Note 63	N	DTLS is not being claimed.
TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Y	
TD0555: NIT Technical Decision for RFC Reference incorrect in TLSS Test	Y	

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0556: NIT Technical Decisions for RFC 5077 question	Y	
TD0563: NIT Technical Decision for Clarification of audit date information	Y	
TD0564: NIT Technical Decision for Vulnerability Analysis Search Criteria	Y	
TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	Y	
TD0570: NIT Technical Decision for Clarification about FIA_AFL.1	Y	
TD0571: NIT Technical Decision for Guidance on how to handle FIA_AFL.1	Y	
TD0572: NIT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Y	
TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Y	
TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	Y	
TD0591: NIT Technical Decision for Virtual TOEs and hypervisors	Y	
TD0592: NIT Technical Decision for Local Storage of Audit Records	Y	
TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server	N	FCS_SSHS_EXT is not being claimed.
TD0632: NIT Technical Decision for Consistency with Time Data for vNDs	N	TOE does not obtain time from the underlying virtualization time
TD0633: NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	N	FCS_IPSEC is not being claimed

Technical Decision	Applicable (Y/N)	Exclusion Rationale (if applicable)
TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters	Y	
TD0636: NIT Technical Decision for Clarification of Public Key User Authentication for SSH	N	FCS_SSHC_EXT is not being claimed.
TD 0638: Technical Decision for Key Pair Generation for Authentication	Y	
TD 0639: NIT Technical Decision for Clarification for NTP MAC Keys	N	NTP is not included in the TOE.
TD 0670: NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	Y	
TD0738: NIT Technical Decision for Link to Allowed-With List	Y	
TD0790: NIT Technical Decision: Clarification Required for testing IPv6	Y	
TD0792: NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	Y	

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.

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). If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform. [TD0591 applied]
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.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.VS_TRUSTED_ADMINISTRATOR	The Security Administrators for the VS are assumed to be trusted and to act in the best interest of security for the organization. This includes not interfering with the correct operation of the device. The Network Device is not expected to be capable of defending against a malicious VS Administrator that actively works to bypass or compromise the security of the device.
A.VS_REGULAR_UPDATES	The VS software is assumed to be updated by the VS Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.VS_ISOLATION	For vNDs, it is assumed that the VS provides, and is configured to provide sufficient isolation between software running in VMs on the same physical platform. Furthermore, it is assumed that the VS adequately protects itself from software running inside VMs on the same physical platform.

ID	Assumption
A.VS_CORRECT_CONFIGURATION	For vNDs, it is assumed that the VS and VMs are correctly configured to support ND functionality implemented in VMs.

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 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 7 – 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 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.

ID	Objectives for the Operational Environment
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.
OE.VM_CONFIGURATION	For vNDs, the Security Administrator ensures that the VS and VMs are configured to • Reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and • Correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting).

5 Security Requirements and Extended Components

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

Table 8 – 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.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_HTTPS_EXT.1	HTTPS Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_TLSC_EXT.1	TLS Client Protocol without Mutual Authentication
FCS_TLSS_EXT.1	TLS Server Protocol
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_MOF.1/Services	Management of Security Functions Behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on security roles
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_TST_EXT.1	TSF Testing

Requirement	Description
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_TRP.1/Admin	Trusted Path

All of the extended requirements in this ST have been drawn from the NDcPP22e. The NDcPP22e defines the following extended requirements and since they are not redefined in this ST the NDcPP22e should be consulted for more information regarding those CC extensions.

- NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP22e:FCS_HTTPS_EXT.1: HTTPS Protocol
- NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP22e:FCS_TLSC_EXT.1: TLS Client Protocol Without Mutual Authentication
- NDcPP22e:FCS_TLSS_EXT.1: TLS Server Protocol Without Mutual Authentication
- NDcPP22e:FIA_PMG_EXT.1: Password Management
- NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
- NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP22e:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests
- NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
- NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps
- NDcPP22e:FPT_TST_EXT.1: TSF testing
- NDcPP22e:FPT_TUD_EXT.1: Trusted update
- NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking

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 <u>underlined</u> 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) All auditable events for the not specified level of audit; and
c) *All administrative actions comprising:*
   - *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
   - *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
   - *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
   - *Resetting passwords (name of related user account shall be logged).*
   - no other actions~~.~~

d) *Specifically defined auditable events listed in Table 9*.

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 9.

Table 9 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	Start-up of the audit function	None
	Shutdown of the audit function	None
	Administrative Login	Name of user account shall be logged if individual user accounts are required for Administrators
	Administrative Logout	
	Changes to TSF data related to configuration changes	In addition to the information that a change occurred, it shall be logged what has been changed
	Generating/import of cryptographic keys	In addition to the action itself, a unique key name or key reference shall be logged
	Changing of cryptographic keys	
	Deleting of cryptographic keys	
	Resetting passwords	Name of related user account shall be logged.
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None

Requirement	Auditable Events	Additional Audit Record Contents
FCS_CKM.1	None	None
FCS_CKM.2	None	None
FCS_CKM.4	None	None
FCS_COP.1/DataEncryption	None	None
FCS_COP.1/SigGen	None	None
FCS_COP.1/Hash	None	None
FCS_COP.1/KeyedHash	None	None
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure
FCS_RBG_EXT.1	None	None
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
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
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None	None
FMT_MOF.1/Functions	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None
FMT_MOF.1/Services	None	None
FMT_MTD.1/CoreData	None	None
FMT_MTD.1/CryptoKeys	None	None
FMT_SMF.1	All management activities of TSF data	None
FMT_SMR.2	None	None
FPT_APW_EXT.1	None	None
FPT_SKP_EXT.1	None	None

Requirement	Auditable Events	Additional Audit Record Contents
FPT_TST_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time (Administrator actuated)	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	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
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 *drop new audit data* when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

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

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
- *ECC schemes using "NIST curves" [*P-256, P-384, P-521*] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4*;].

]

5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

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

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

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

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

5.2.2.3 FCS_CKM.4 Cryptographic Key Destruction

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

- *For plaintext keys in volatile storage, the destruction shall be executed by a [*destruction of reference to the key directly followed by a request for garbage collection*];*

- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [*
  - instructs a part of the TSF to destroy the abstraction that represents the key]

that meets the following: *No Standard*

5.2.2.4 FCS_COP.1/DataEncryption Cryptographic 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] *mode* and cryptographic key sizes [128 bits, 256 bits] that meet the following: *AES as specified in ISO 18033-3,* [CBC as specified in ISO 10116, GCM as specified in ISO 19772].

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

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

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

]
that meet the following: [

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

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

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

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

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

5.2.2.8 FCS_HTTPS_EXT.1 HTTPS Protocol

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

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

FCS_HTTPS_EXT.1.3
If a peer certificate is presented, the TSF shall [not establish the connection] if the peer certificate is deemed invalid.

5.2.2.9 FCS_RBG_EXT.1 Random Bit Generation

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

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[2] platform-based noise sources*] with a minimum of [256 bits] of entropy at least equal to the greatest

security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

5.2.2.10 FCS_TLSC_EXT.1 TLS Client Protocol without Mutual Authentication

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

[

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC5246
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

] and no other ciphersuites.

FCS_TLSC_EXT.1.2
The TSF shall verify that the presented identifier matches [*the reference identifier per RFC 6125 section 6, and no other attribute types*].

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

- Not implement any administrator override mechanism].

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

5.2.2.11 FCS_TLSS_EXT.1 TLS Sever Protocol Without Mutual Authentication

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

[

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC5246
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

] and no other ciphersuites.

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

FCS_TLSS_EXT.1.3
The TSF shall perform key establishment for TLS using [RSA with key size [2048 bits, 3072 bits, 4096 bits], ECDHE curves [secp256r1, secp384r1, secp521r1] *and no other curves*].

FCS_TLSS_EXT.1.4
The TSF shall support [session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), session resumption based on session tickets according to RFC 5077].

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 *[1-15]* 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 [*6*] and [*15*] 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.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1

The TSF shall provide a local [password-based, *remote password-based authentication using LDAP/S*] authentication mechanism to perform local administrative user authentication.

5.2.3.5 FIA_UAU.7.Protected Authentication Feedback

FIA_UAU.7.1

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

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 [*the Online Certificate Status Protocol (OCSP) as specified in RFC 6960*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
    - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
    - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsagefield.*
    - *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 [HTTPS, TLS] and [no additional uses].

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

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, Organizational Unit, Country].

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

5.2.4 Security Management (FMT)

5.2.4.1 FMT_MOF.1/Functions Management of Security Functions Behaviour

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

5.2.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions 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_MOF.1/Services Management of Security Functions Behaviour

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

5.2.4.4 FMT_MTD.1/CoreData Management of TSF Data

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

5.2.4.5 FMT_MTD.1/CryptoKeys Management of TSF Data

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

5.2.4.6 FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [*hash comparison*] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [
    - *Ability to start and stop services;*
    - *Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);*
    - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
    - *Ability to manage the cryptographic keys;*
    - *Ability to configure the cryptographic functionality;*
    - *Ability to set the time which is used for time-stamps;*
    - *Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;*
    - *Ability to import X.509v3 certificates to the TOE's trust store;*
    - *No other capabilities].*

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 Protection of the TSF (FPT)

5.2.5.1 FTP_APW_EXT.1 Protection of Administrator Passwords

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

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

5.2.5.2 FPT_SKP_EXT.1 Protection of TSF Data (for reading 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.

5.2.5.3 FPT_STM_EXT.1 Reliable Time Stamps

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

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

5.2.5.4 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [*during initial start-up (on power on), at the request of the authorized user]* to demonstrate the correct operation of the TSF:

- SHA256 checksum of the code comprising the application;
- SHA256 checksum of the supporting OS packages

5.2.5.5 FPT_TUD_EXT.1 Trusted Update

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

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

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

5.2.6 TOE Access (FTA)

5.2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1
The TSF Shall, for local interactive sessions, [

- terminate the session]

after a Security Administrator-specified time period of inactivity.

5.2.6.2 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.*

5.2.6.3 FTA_SSL.4 User-initiated Termination

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

5.2.6.4 FTA_TAB.1 Default TOE Access Banners

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

5.2.7 Trusted Path/Channels (FTP)

5.2.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

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

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

FTP_ITC.1.3
The TSF shall initiate communication via the trusted channel for *[*audit log forwarding, authentication, mail notification*]*.

5.2.7.2 FTP_TRP.1/Admin Trusted Path

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

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

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

5.3 TOE 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 EPs/Modules/Packages, which is/are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 10.

Table 10 – 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 HYCU, Inc. to satisfy the assurance requirements. The following table lists the details.

Table 11 – 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	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_CMS.1	

SAR Component	How the SAR will be met
ATE_IND.1	Vendor will provide the TOE for testing.
AVA_VAN.1	Vendor will provide the TOE for testing. Vendor will provide a document identifying the list of software and hardware components.

6 TOE Summary Specification

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

Table 12 – TOE Summary Specification SFR Description

Requirement	TSS Description
FAU_GEN.1	The TOE generates a comprehensive set of audit logs that identify specific TOE operation whenever an auditable event occurs. Auditable events are specified in Table 9 – Security Functional Requirements and Auditable Events. Each of the events specified in the audit records is in enough detail to identify the user for which the event is associated, when the event occurred, where the event occurred, the outcome of the event and the type of event that occurred. Administrative tasks of generating and deleting cryptographic keys identify the key name. The TOE generates audit records when specific events occur. The audit events recorded are comprehensive and all events are specified in Table 9 – Security Functional Requirements and Auditable Events. Each recorded event includes sufficient detail to identify the user associated with the event, when the event occurred, where the event occurred, the outcome of the event, and the type of event. The key name is identified for administrative tasks of generating, changing and deleting cryptographic keys.
FAU_STG_EXT.1	Audit events are stored locally and are also sent to an external audit server as they are created. TLS is used to provide a trusted communication channel with the audit server. Data is stored locally in a database which also contains other system information, consequently there is no set limit on the local audit log storage. An administrator can configure the database to purge events older than a specified date. The TOE sends audit records to the audit server in real-time. If communication with the audit server fails, events are stored locally and when the connection is restored all stored audit records will be transmitted to the remote audit server. If local storage space is exhausted new audit records are dropped. Other than an administrator being able to clear the local audit records there is no provision to modify the records. An audit record is generated when the audit log is cleared. The amount of audit data is not explicitly bound by size, but it is limited by disk space available to the database, which is in turn limited by the space available on the data disk of the HYCU appliance. When the storage data is exhausted, all database activities cease until space is made available. Until then, new audit records are dropped. The TOE is standalone and not distributed so all audit data is stored locally.
FCS_CKM.1	The TOE supports RSA and ECC cryptographic key generation schemes which include RSA 2048-bit and ECC P-256, ECC P-384, ECC P-521s. These are detailed in FCS_CKM.1. RSA and ECC are used for TLSC and TLSS.

Requirement	TSS Description
FCS_CKM.2	In agreement with the key generation schemes in FCS_CKM.1.1 the RSA-based and Elliptic curve-based key establishment schemes are supported as detailed in FCS_CKM.2. Both RSA-based and Elliptic curve-based key generation schemes are used for TLSC and TLSS.
FCS_CKM.4	The TOE stores plaintext keys in volatile and non-volatile storage. The TOE satisfies all requirements for destruction of keys and CSPs as specified in FCS_CKM.4. Please refer to Table 14 – Key Storage and Deletion. There are no keys stored as non-plaintext. The TOE uses the following plaintext keys: 1. Private key for TLS, used by the HTTPS server to serve remote administration API and user interface 2. X.509 certificates for TLS, used by: o the HTTPS server to serve remote administration API and user interface o the HTTP clients to establish trusted roots 3. Password for authentication to TOE and to external systems (LDAPS, SMTP, Webhook). Ad 1. Private key Private key is stored in non-volatile storage: - Authoritative copy is stored in a PostgreSQL database internal to the TOE (database ‘cfgdb’, table ‘certificate’), in an encrypted form using PBKDF2 is used to derive a key, which is then encrypted using AES-CBC before stored in the database. Certificate table is stored as pages on the disk under /hycudata/opt/grizzly/data. Key destruction is handled by DELETE SQL statement. On-disk data is garbage-collected and legible for overwrite after autovacuum daemon executes the VACUUM SQL statement. - Private key is also stored on the filesystem (in /etc/pki/tls/private/hycussl-*) for use by the HTTP/S server component. Key is rendered into the file on every Java service startup. Since the TOE mandates use of HTTP/S, the private key is always present and is only deleted when switching to a different key. If multiple listeners are configured (e.g. multiple network interfaces), each listener may be configured with a separate private key. On-disk key deletion is done using \`rm\` system command. Private key is also stored in volatile storage: - HTTPS server loads the private key into volatile storage to accept TLS connections. Private key remains loaded in volatile storage for the duration of the mod_ssl module lifetime. When module lifetime ends (on HTTP server shutdown or reload), volatile storage is freed using standard free() call. Underlying OS ensures pages are zeroized before next use.

Requirement	TSS Description
	- Java application loads the private keys into volatile storage when registering new keys into memory, and when rendering keys on the filesystem. After use, references are released, and memory reclaimed by garbage collection. On service shutdown, pages used by the java application are freed by the OS, which ensures pages are zeroized before next use. Ad 2. X.509 certificates Certificates are stored in non-volatile storage: - Authoritative copy is stored in a PostgreSQL database internal to the TOE (database ‘cfgdb’, table ‘certificate’), in plaintext form, since the data is public. Key destruction is handled by DELETE SQL statement. On-disk data is garbage-collected and legible for overwrite after autovacuum daemon executes the VACUUM SQL statement. - Server certificate is also stored on the filesystem (in /etc/pki/tls/certs/hycussl-*) for use by the HTTP/S server component. Certificate is rendered into the file on every Java service startup. Since the TOE mandates use of HTTP/S, the certificate is always present and is only deleted when switching to a different key. If multiple listeners are configured (e.g. multiple network interfaces), each listener may be configured with a separate private key. On-disk certificate deletion is done using \`rm\` system command. Certificates are also stored in volatile storage: - HTTPS server loads the certificate into volatile storage to accept TLS connections. Certificate remains loaded in volatile storage for the duration of the mod_ssl module lifetime. When module lifetime ends (on HTTP server shutdown or reload), volatile storage is freed using standard free() call. Underlying OS ensures pages are zeroized before next use. - Java application loads the certificate from database into volatile storage on startup, or when registering new or removing old certificates. These certificates are used by the TLS client to establish trusted roots during handshake. Certificates are loaded into memory in BouncyCastle’s BCFKS FIPS-validated trust store. When certificates change, in-memory keystore is rebuilt, and references to old keystore are released, and memory reclaimed by automatic garbage collection. On service shutdown, pages used by the java application are freed by the OS, which ensures pages are zeroized before next use. Ad 2. Passwords Passwords are stored in non-volatile storage: - Authoritative copy of SMTP and Webhook password is stored in a database internal to the TOE (SMTP: database ‘cfgdb’, table ‘smtp’; Webhook: database \`grizzly\`, table ‘webhook\`), in an encrypted form using PBKDF2 is used to derive a key, which is

Requirement	TSS Description
	then encrypted using AES-CBC before being stored in the database. Passwords are stored in volatile storage: - Passwords for Active Directory authentication (LDAP bind) are loaded from non-volatile on-demand (e.g. during login). After use, references are released, and memory reclaimed by garbage collection. - Password for SMTP (if set) is loaded from non-volatile storage on service startup or SMTP configuration change and remains loaded for the duration of the service (or until SMTP configuration change). On SMTP configuration change, references are released, and memory reclaimed by garbage collection. On service shutdown, pages used by the java application are freed by the OS, which ensures pages are zeroized before next use.
FCS_COP.1/DataEncryption	The TOE supports AES encryption and decryption conforming to CBC and GCM as specified in ISO 18033-3, ISO 10116, and ISO 19772. The AES key size supported is 128 and 256 bits.
FCS_COP.1/Hash	Cryptographic hashing supports TLS and HTTPS using SHA-1, SHA-256, or SHA-384 with message digest sizes of 160, 256, or 384.
FCS_COP.1/KeyedHash	Keyed-hash message authentication supports TLS and HTTPS using HMAC-SHA-1, HMAC-SHA-256, or HMAC-SHA-384 with cryptographic key sizes of 160, 256, or 384 bits, message digest sizes of 160, 256, or 384 bits. The block size for HMAC-SHA-1 is 64 bytes. For HMAC-SHA-256 and HMAC-SHA-384 the block size is 128 bytes.
FCS_COP.1/SigGen	The TOE provides cryptographic signature generation and verification services in accordance with the following cryptographic algorithms: • RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048, 3072 and 4096 bits] according to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3 • Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256, 384, or 521 bits] according to 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
FCS_HTTPS_EXT.1	The TOE supports remote management of the TOE over an HTTPS connection using TLS. In this scenario, the TOE acts as a server. This protocol is used to provide an administrator with access to the TOE. The HTTPS protocol complies with RFC 2818.
FCS_RBG_EXT.1	The following DRBG types are supported: • HYCU Java cryptographic library supports Hash, HMAC and CTR DRBG (AES). • HYCU Native cryptographic library supports CTR DRBG (AES).

Requirement	TSS Description
	The deterministic RBG is seeded by 2 entropy sources that accumulates entropy from add_interrupt_randomness() (i.e the interrupt noise source) and add_disk_randomness() (i.e. the disk noise sources). There is a minimum of 256 bits of entropy at least equal to the greatest security strength possible according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate
FCS_TLSC_EXT.1	This applies to communication between the TOE and an authentication server, a mail server, and an audit server. This channel requires the use of TLS 1.2 and this is enforced by the TOE. The supported cipher suites are the following: • TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 • TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246 • TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246 • TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288 • TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 The TSF ensures that the presented reference identifier conforms to RFC 6125 section 6 and if FQDN in SAN matches. The TSF will not establish a trusted channel if the server certificate is invalid and there is no provision for this to be overridden. The TSF will use the algorithm specified by the server provided that it is one of the ones listed above. The algorithm is not configurable. Wildcards are supported by the TOE. Certificate pinning is not supported. The TSF presents the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups in the client hello: secp256r1, secp384r1, and secp521r1. These curves are supported by default and do not need to be configured by an administrator.
FCS_TLSS_EXT.1	This applies to communication between the TOE and the administrator workstation (web browser). The use of TSL 1.2 is mandated by the TSF and connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and TLS 1.1 are denied. The supported ciphersuites are the following:

Requirement	TSS Description
	• TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 • TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246 • TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246 • TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288 • TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 Key establishment is performed using RSA with key sizes of 2048 bits, 3072 bits, or 4096 bits and ECDHE curves secp256r1, secp384r1, or secp521r1. ECDHE and DHE ciphers utilize standard DH parameters: - RFC 2049, section 6.2 - RFC 3526, sections 3 to 7 Parameters are selected based on authentication strength (or key size for RSA keys) configured via CC-Mode. Session resumption based on session IDs is supported conforming to RFC4346 and RFC5346. Session resumption tickets adhere to the structural format described in Section 4 of RFC 5077. Session tickets are protected according to recommendations outlined in Section 4 of RFC 5077: - encrypted with AES CBC with 128-bit key, and - MAC calculated using HMAC-SHA-256. The TOE does not support DTLS.
FIA_AFL.1	An administrator can configure the maximum number of failed attempts using the CLI interface. The configurable range is between 1-15 attempts with the default being 3 attempts. When a user account has sequentially failed authentication for the configured number of times, the account will be locked for the configured period of time or until a local administrator manually unlocks the account. All failed attempts and lockouts are tracked by the TOE audit log. The TOE will always allow a user to authenticate using the local console port, even if the user account is locked. This behavior is not configurable. There is a single console/OS

Requirement	TSS Description
	account ('hycu') which is meant only for specific operations, which could include recovery from any kind of application or system failure.
FIA_PMG_EXT.1	The TOE supports passwords that can be composed of any combination of upper and lower case letters, numbers, and the following special characters: ["!", "@", "#", "$", "%", "^", "&", "*", "(", ")"] Minimum password length is configurable to between [6] and [15] characters.
FIA_UIA_EXT.1	The TOE supports two login methods. One is the local console and the other is the web GUI (HTTPS/TLS). For both methods users are presented with a login banner prior to login. Without a successful login a user can only see the login banner and login screen. In order to login a user must provide a username and password.
FIA_X509_EXT.1/Rev	Validity of certificates takes place during authentication. Revocation checking is done during authentication on all certificates in the provided chain using OCSP. If the OCSP responder certificate does not contain the OCSP signing bit extendedKeyUsage, the connection will fail. All TLS certificates used to authenticate to the TOE must contain the Server Authentication extendedKeyUsage bit.
FIA_X509_EXT.2	TOE uses the server certificate configured by the user. Configuration is described in the HYCU user guide. TOE will not establish a trusted channel if the certificate validity check fails for any reason. This behavior is not configurable. Trusted channels include a TLS connection to an audit, authentication and SMTP server.
FIA_X509_EXT.3	When generating a certificate request the TSF provides the public key, common name, organization, organizational unit, and country in the request. There is no device-specific information provided in the CSR.
FMT_MOF.1/Functions	The TOE restricts the ability to configure the transmission of audit records to an external audit server to the security administrator. No other users have the ability to modify the behavior of the transmission of audit data to an external IT entity.
FMT_MOF.1/ManualUpdate	The TOE can only be updated manually by a security administrator, and this must be performed from the console.
FMT_MOF.1/Services	A security administrator can enable communication to an external audit server via TLS. A security administrator can also disable audit log sending by clearing the notification configuration. In addition to the audit server service, the administrator is also able to start an AD server service and an SMTP server service. The admin can start these services by configuring settings via the web GUI. The admin can also stop these services by deleting their configured settings in the web GUI.
FMT_MTD.1/CoreData	All TOE users are required to login and there is no functionality provided prior to authentication other than displaying the TOE banner on every login interface. All abilities to manipulate TSF data are handled by an administrator and no other user accounts. The TOE restricts the ability to manage the trust store only to the administrator. No other user accounts can manage this functionality.

Requirement	TSS Description
FMT_MTD.1/CryptoKeys	The generation, importing, and deletion of cryptographic keys is restricted to the security administrator. HYCU has a concept of user groups. Users in one user group cannot see entities owned by another user group. “Infrastructure group” is a built-in administrative group that can see entities from other user groups. Administrator role in “Infrastructure group” confers higher privileges than administrator role in a user group. Security administrator is able to control the following cryptographic keys: • X.509 certificates for HTTP/S server ○ Generate, import, delete • X.509 certificates used for trusted roots ○ Import, delete • Passwords for access to LDAP and SMTP servers ○ Create, modify, delete (by disabling LDAP/SMTP integration) • Passwords for local HYCU appliance users ○ Create, modify, delete (by removing user) Procedures for controlling these cryptographic keys are described in the guidance materials.
FMT_SMF.1	The TOE can be managed via the web GUI and local console. Management activities that can be performed though the web GUI include the following: - start and stop services - update the TOE - modify the behavior of the transmission of audit data to an external IT entity - Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full) - manage the cryptographic keys - configure the cryptographic functionality - manage the TOE's trust store and designate X509.v3 certificates as trust anchors - import X.509v3 certificates to the TOE's trust store The following management activities can be performed using the console: - set the time which is used for time-stamps (console only)) - Configuring access banners - Ability to configure the session inactivity time before session termination or locking - Ability to configure the authentication failure parameters for FIA_AFL.1

Requirement	TSS Description
FMT_SMR.2	The administrator can assign users to groups and roles. These groups consist of the infrastructure group and the self-service group. In addition to the group assignment users must also be assigned a role. Access to TSF functions is limited to users assigned the administrator role. Infrastructure group administrators can manage users, create/edit/delete self-service groups, add/remove users from groups, activate/deactivate users and self-service groups, and set owners of VMs and file shares. A self-service group administrator can add/remove users from groups. Roles and Restrictions: - Administrator ○ In user group: can administer membership. ○ In “Infrastructure group”: can configure appliance. - Backup operator – can define policies, assign them to entities and trigger backup and restore operations. - Restore operator – can trigger restore operations. - Backup and restore operator – combine privileges of both backup and restore operators - Viewer – can view (but not modify) entities. TSF data can only be managed by security administrators via the console or web GUI.
FPT_APW_EXT.1	Passwords are never displayed or stored in plaintext form. Passwords are obscured during login, they are not stored in the audit log, and they are not stored in plaintext form. Passwords are stored in the database in an encrypted form - PBKDF2 is used to derive a key from the password, which is then encrypted using AES-CBC before stored in the database. Passwords are only decoded on retrieval by the internal workings of the appliance, and the values are not available via any user-facing API.
FPT_SKP_EXT.1	Symmetric, private and pre-shared keys (including passwords needed for remote login to systems such as LDAPS/AD and SMTP) are stored in the database in an encrypted form. Public keys/certificates are public and hence not encrypted. Details of storage and encryption for private keys, certificates and passwords is described in FCS_CKM.4 TSS. There are mechanisms in place to prevent viewing key information through an interface designed specifically for that purpose.
FPT_STM_EXT.1	The TOE provides reliable time stamps security audit functionality, administrative session inactivity, and cryptographic functions. The system time can only be changed by a security administrator. There is no possible delay because the time is not provided by the underlying virtual system. The time is manually set by the admin.

Requirement	TSS Description
FPT_TST_EXT.1	The TOE performs the self-tests on startup and on reload. The self-tests can also be initiated from the console, as described in the guidance documentation. Self-tests check the following: - the integrity of the application code, by calculating SHA256 checksum of all the application files and comparing them against checksums stored in a file created at the time of the build. - the integrity of the openssl, java and kernel operating system packages (using native RPM -v). If the TOE starts, we can infer the following: • The HYCU application server has started successfully. • Database has been started and schema migrations have completed successfully. • The reverse proxy is running and configured correctly. Self-test that run before server is started ensuring that the state of the application server and cryptographically relevant parts of the OS have not been tampered with and match the state at release time. As mandated by FIPS-140-2, self-test is performed on cryptographic library initialization at first operation involving the cryptographic library. If this self-test passes, operations continue normally. If this self-test fails, the self-test failure reason is logged, and TOE startup is prevented. The native cryptographic library (OpenSSL) self-test failure prevents Apache httpd startup. Java cryptographic library (BouncyCastle) self-test failure prevents HYCU application (grizzly) startup.
FPT_TUD_EXT.1	The administrator can determine the current TOE version from either the console or web GUI and they can install a new version using the console or web GUI after authentication as an administrator. The administrator is responsible to verify the hash of the update prior to installation. The hash is made available by HYCU when the TOE update is provided. When an update is performed it takes effect immediately. None of the update procedures are automated. If the hash verification fails, the TOE will not initialize and will have to be reverted to the previous version. If the hash verification succeeds, the TOE will proceed with the update process and the version will change.
FTA_SSL.3	Inactive remote user and administrator sessions are terminated after an administrator configured time interval. Local and remote session timeout can be configured using the console interface.
FTA_SSL.4	Remote administrator sessions and local administrator sessions can be terminated by the administrator who is logged in to the session. The session is terminated by the admin executing the logout command on the Web GUI or CLI interface.
FTA_SSL_EXT.1	Inactive remote administrator sessions and console sessions are terminated by the TOE after an administrator defined period of time.

Requirement	TSS Description
	Remote administrator session timeout is configurable via /opt/grizzly/config.properties variable api.session.expiration.minutes, with default set to 15 (minutes). Console session timeout is configurable via /etc/profile.d/bash-autologout.sh, with TMOUT environment variable value by default set to 600 (seconds, or 10 minutes).
FTA_TAB.1	The TOE provides separate login banners for the web GUI and console. The guidance documentation requires that during initial configuration of the TOE, an appropriate advisory notice and consent warning message is configured for both remote and local methods of access. The message configured is shown on both login methods.
FTP_ITC.1	The TOE uses trusted channels to protect communication with an external audit server, authentication server, and SMTP server. HTTPS POST webhooks are used for communication with an external audit server. Basic Authentication (username/password) is used for HTTPS connections via the web GUI for the TOE. Authentication via certificates is used for connections to the audit, SMTP and authentication server. LDAP/S (TLS 1.2) is used to protect communication with the authentication server. The TOE verifies that the LDAP server hostname matches the DNS entry specified in the Subject Alternative Name (SAN) extension of the LDAP server's certificate. SMTP/S (TLS 1.2) is used to protect communication with the SMTP server with the TOE acting as the client.
FTP_TRP.1/Admin	HTTPS (TLS 1.2) is used to secure remote administration. This is the only remote administration method that is available.

6.1 CAVP Algorithm Certificate Details

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

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3	HYCU Java Cryptographic Library	RSA KeyGen	#A2933
	ECC schemes using “NIST curves” [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4	HYCU Java Cryptographic Library	ECDSA KeyGen	#A2933

FCS_CKM.2	RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1”	HYCU Java Cryptographic Library	None: CCTL tested as per the PP/SD Evaluation Activities	Lab Evaluated
	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”	HYCU Java Cryptographic Library	KAS-ECC-SSC	#A2933
FCS_COP.1/DataEncryption	AES used in [CBC, GCM] mode and cryptographic key sizes [128 bits, 256 bits]	HYCU Java Cryptographic Library	AES-CBC AES-GCM	#A2933
FCS_COP.1/SigGen	For RSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	HYCU Java Cryptographic Library	RSA-SigGen	#A2933
	For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4	HYCU Java Cryptographic Library	ECDSA-SigGen	#A2933
FCS_COP.1/Hash	[SHA-1, SHA-256, SHA-384] and message digest sizes [160, 256, 384] bits	HYCU Java Cryptographic Library	SHA-1 SHA2-256 SHA2-384	#A2933
FCS_COP.1/KeyedHash	[HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384] and cryptographic key sizes [160, 256, and 384 bits] and message digest sizes [160, 256, 384] bits	HYCU Java Cryptographic Library	HMAC-SHA-1 HMAC-SHA2-256	#A2933

			HMAC-SHA2-384	
FCS_RBG_EXT.1	Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)	HYCU Java Cryptographic Library	Hash DRBG HMAC DRBG Counter DRBG	#A2933

Table 13 – CAVP Algorithm Certificate References

6.2 Cryptographic Key Destruction

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

Table 14 – Key Storage and Deletion

Keys/CSPs	Purpose	Storage Location	Deletion Method
Private key for TLS	used by the HTTPS server to serve remote administration API and user interface. Key is generated when the administrator enables GUI access via HTTPS.	Non-volatile storage: Authoritative copy is stored in a PostgreSQL database internal to the TOE (database 'cfgdb', table 'certificate'), in an encrypted form using PBKDF2 is used to derive a key, which is then encrypted using AES-CBC before stored in the database. Certificate table is stored as pages on the disk under /hycudata/opt/grizzly/data. Private key is also stored on the filesystem (in /etc/pki/tls/private/hycussl-*) for use by the HTTP/S server component. Key is rendered into the file on every Java service startup. Volatile storage: HTTPS server loads the private key into volatile storage to accept TLS connections. Private key	Non-volatile storage: Key destruction is handled by DELETE SQL statement. On-disk data is garbage-collected and legible for overwrite after autovacuum daemon executes the VACUUM SQL statement. Since the TOE mandates use of HTTP/S, the private key is always present and is only deleted when switching to a different key. If multiple listeners are configured (e.g. multiple network interfaces), each listener may be configured with a separate private key. On-disk key deletion is done using \`rm\` system command. Volatile Storage:

Keys/CSPs	Purpose	Storage Location	Deletion Method
		remains loaded in volatile storage for the duration of the mod_ssl module lifetime. Java application loads the private keys into volatile storage when registering new keys into memory, and when rendering keys on the filesystem.	When module lifetime ends (on HTTP server shutdown or reload), volatile storage is freed using standard free() call. Underlying OS ensures pages are zeroized before next use. After use, references are released, and memory reclaimed by garbage collection. On service shutdown, pages used by the java application are freed by the OS, which ensures pages are zeroized before next use.
Passwords	Used to log accounts in. Is used for SMTP and AD connections on the TOE. Key is generated when the administrator creates a new user account with a set password.	Non-volatile storage: Authoritative copy of SMTP and Webhook password is stored in a database internal to the TOE (SMTP: database 'cfgdb', table 'smtp'; Webhook: database `grizzly`, table 'webhook`), in an encrypted form using PBKDF2 is used to derive a key, which is then encrypted using AES-CBC before being stored in the database. Volatile Storage: Passwords for Active Directory authentication (LDAP bind) are loaded from non-volatile on-demand (e.g. during login). After use, references are released, and memory reclaimed by garbage collection. Password for SMTP (if set) is loaded from non-volatile storage on service startup or SMTP configuration change, and remains loaded for the duration of the	Volatile storage: On SMTP configuration change, references are released, and memory reclaimed by garbage collection. On service shutdown, pages used by the java application are freed by the OS, which ensures pages are zeroized before next use.

Keys/CSPs	Purpose	Storage Location	Deletion Method
		service (or until SMTP configuration change).	
Public Keys	X.509 certificates for TLS, used by: • the HTTPS server to serve remote administration API and user interface • the HTTP clients to establish trusted roots Keys are generated when the TOE generates a CSR or HTTPS certificate for use by the TOE GUI.	Non-volatile storage: -Authoritative copy is stored in a PostgreSQL database internal to the TOE (database 'cfgdb', table 'certificate'), in plaintext form, since the data is public. -Server certificate is also stored on the filesystem (in /etc/pki/tls/certs/hycussl-*) for use by the HTTP/S server component. Certificate is rendered into the file on every Java service startup. Since the TOE mandates use of HTTP/S, the certificate is always present and is only deleted when switching to a different key. If multiple listeners are configured (e.g. multiple network interfaces), each listener may be configured with a separate private key. Volatile Storage: -HTTPS server loads the certificate into volatile storage to accept TLS connections. Certificate remains loaded in volatile storage for the duration of the mod_ssl module lifetime. When module lifetime ends (on HTTP server shutdown or reload), volatile storage is freed using standard free() call. -Java application loads the certificate from database into volatile storage on startup, or when registering new or removing old certificates. These certificates are used by the TLS client to establish	Non-volatile storage: Key destruction is handled by DELETE SQL statement. On-disk data is garbage-collected and legible for overwrite after autovacuum daemon executes the VACUUM SQL statement. On-disk certificate deletion is done using `rm` system command. Volatile storage: When module lifetime ends (on HTTP server shutdown or reload), volatile storage is freed using standard free() call. Underlying OS ensures pages are zeroized before next use. When certificates change, in-memory keystore is rebuilt, and references to old keystore are released, and memory reclaimed by automatic garbage collection. On service shutdown, pages used by the java application are freed by the OS, which ensures pages are zeroized before next use.

Keys/CSPs	Purpose	Storage Location	Deletion Method
		trusted roots during handshake. Certificates are loaded into memory in BouncyCastle's BCFKS FIPS-validated trust store.	

7 Acronym Table

Acronyms should be included as an Appendix in each document.

Table 15 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CC	Common Criteria
CRL	Certificate Revocation List
DRBG	Deterministic Random Number Generator
ECDHE	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Diffie-Hellman Ephemeral
EP	Extended Package
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
LDAP	Lightweight Directory Access Protocol
NDcPP	Network Device Collaborative Protection Profile
NIAP	Nation Information Assurance Partnership
OCSP	Online Certificate Status Protocol
PP	Protection Profile
RSA	Rivest, Shamir & Adleman
SFR	Security Functional Requirement
SLO	Service Level Objective
ST	Security Target
TD	Technical Decision
TOE	Target of Evaluation
TLS	Transport Layer Security
TSF	TOE Security Functionality
TSS	TOE Summary Specification