NETSCOUT

NETSCOUT Omnis Cyber Intelligence with CyberStream v6.3.4

Security Target

Version 1.12

October 2024

Document prepared by

Lightship Security

www.lightshipsec.com

Document History

Version	Date	Description
0.5	04 August 2023	OR3 addressed
1.0	13 September 2023	Addressing CB OR, published for evaluation.
1.1	05 October 2023	Addressing CB OR.
1.2	15 November 2023	Include iDRAC in excluded interfaces
1.3	26 January 2024	Addressing Evaluator Comments
1.4	12 February 2024	Addressing Evaluator Comments
1.5	26 February 2024	Addressing OR05
1.6	08 March 2024	Linked Eval Updates OR18
1.7	19 Apr 2024	Linked Eval Updates OR19
1.8	15 May 2024	Addressing Evaluator Comments
1.9	21 May 2024	Addressing Evaluator Comments for AGD
1.10	27 Jun 2024	Addressing Evaluator Comments for AGD
1.11	08 Aug 2024	Addressing Comments for AGD
1.12	17 Oct 2024	Updating TD applicability

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the NETSCOUT Omnis Cyber Intelligence with CyberStream v6.3.4 distributed Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 The Omnis Cyber Intelligence (OCI) component provides centralized management and reporting capabilities for enterprise cybersecurity and service monitoring. One or more CyberStream components are deployed throughout an enterprise network to enable monitoring.

1.2 Identification

Table 1: Evaluation Identifiers

Target of Evaluation	NETSCOUT Omnis Cyber Intelligence with CyberStream v6.3.4 Builds: • OCI: v6.3.4 build 2345 • CyberStream: v6.3.4 build 2048 Patches: OCI: • ATSFX64_34W_54L_53_90058_02zg_208 • nG1_6x-STIG-10JAN2022_v1
Security Target	NETSCOUT Omnis Cyber Intelligence with CyberStream v6.3.4 Security Target, v1.12

1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) CC version 3.1 revision 5

b) CC Part 2 extended

c) CC Part 3 conformant

d) collaborative Protection Profile for Network Devices, v2.2E (NDcPP)

e) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name	Rationale if n/a
TD0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	
TD0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	FCS_NTP_EXT.1 not claimed.

TD #	Name	Rationale if n/a
TD0536	NIT Technical Decision for Update Verification Inconsistency	
TD0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	
TD0546	NIT Technical Decision for DTLS - clarification of Application Note 63	DTLS not claimed
TD0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	
TD0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	
TD0556	NIT Technical Decision for RFC 5077 question	
TD0563	NiT Technical Decision for Clarification of audit date information	
TD0564	NiT Technical Decision for Vulnerability Analysis Search Criteria	
TD0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	
TD0570	NiT Technical Decision for Clarification about FIA_AFL.1	
TD0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1	
TD0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	
TD0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	
TD0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	
TD0591	NIT Technical Decision for Virtual TOEs and hypervisors	Not a virtual Network Device
TD0592	NIT Technical Decision for Local Storage of Audit Records	
TD0631	NIT Technical Decision for Clarification of public key authentication for SSH Server	
TD0632	NIT Technical Decision for Consistency with Time Data for vNDs	Not a virtual Network Device

TD #	Name	Rationale if n/a
TD0635	NIT Technical Decision for TLS Server and Key Agreement Parameters	
TD0636	NIT Technical Decision for Clarification of Public Key User Authentication for SSH	
TD0638	NIT Technical Decision for Key Pair Generation for Authentication	
TD0639	NIT Technical Decision for Clarification for NTP MAC Keys	FCS_NTP_EXT.1 not claimed.
TD0670	NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	
TD0738	NIT Technical Decision for Link to Allowed-With List	
TD0790	NIT Technical Decision: Clarification Required for testing IPv6	
TD0792	NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	
TD0800	Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	IPSEC not claimed

1.4 Terminology

Table 3: Terminology

Term	Definition
AA	Authenticator Address
CC	Common Criteria
EAL	Evaluation Assurance Level
LAN	Local Area Network
NDcPP	collaborative Protection Profile for Network Devices
NDcPP-SD	collaborative Protection Profile for Network Devices Supporting Document
OCI	Omnis Cyber Intelligence
PP	Protection Profile

Term	Definition
TOE	Target of Evaluation
TSF	TOE Security Functionality

2 TOE Description

2.1 Type

4 The TOE is a distributed network device that consists of an OCI management platform and one or more CyberStream appliances.

2.2 Usage

2.2.1 Deployment

5 Figure 1 shows an example deployment of the TOE.

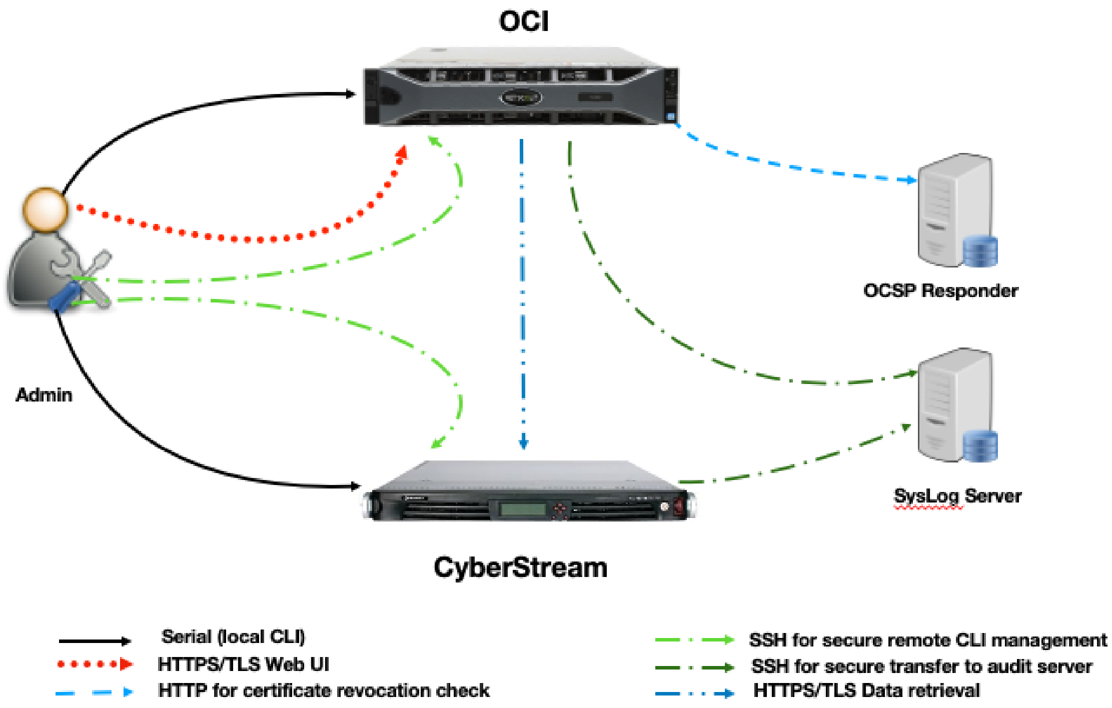
OCI
OCSP Responder
Admin
SysLog Server
CyberStream
Serial (local CLI)
HTTPS/TLS Web UI
HTTP for certificate revocation check
SSH for secure remote CLI management
SSH for secure transfer to audit server
HTTPS/TLS Data retrieval

Figure 1: Example TOE deployment

2.2.2 Interfaces

6 The TOE interfaces are as follows:

a) **CLI.** Administrative CLI via direct serial connection or remote SSH.

b) **GUI.** Administrative OCI Web GUI via HTTPS.

c) **Syslog.** Transmission of logs to the syslog server via SSH.

d) **OCSP Responder.** X.509v3 certificate revocation checking via OCSP.

7 **Note:** NETSCOUT REST APIs do not provide TOE security administration capabilities and are not functionally tested by the Common Criteria evaluation.

2.3 Security Functions/Logical Scope

8 The TOE provides the following security functions:

a) **Protected Communications.** The TOE protects the integrity and confidentiality of communications as noted in section 2.2.2 above.

b) **Secure Administration.** The TOE enables secure management of its security functions, including:

i) Administrator authentication with passwords

ii) Configurable password policies

iii) Access Control

iv) Access banners

v) Management of critical security functions and data

vi) Protection of cryptographic keys and passwords

c) **Trusted Update.** The TOE ensures the authenticity and integrity of software updates through the use of a published hash mechanism.

d) **System Monitoring.** The TOE generates logs of security relevant events. The TOE stores logs locally and is capable of sending log events to a remote audit server.

e) **Self-Test.** The TOE performs a suite of self-tests to ensure the correct operation and enforcement of its security functions.

f) **Cryptographic Operations.** The TOE implements validated cryptographic algorithms.

9 Table 4 lists the relevant CAVP Certificates.

Table 4: CAVP Certificates

Algorithm Capabilities	Certificate
AES-CBC AES-CTR AES-GCM ECDSA KeyGen (186-4) ECDSA SigGen (186-4) ECDSA SigVer (186-4) SHA1 SHA-256/384/512 HMAC-SHA-256/384/512 Counter DRBG	A3740 (for OCI) A3739 (for OCI + CyberStream) A5165 (for OCI)
KAS-ECC-SSC Sp800-56Ar3 KAS-FFC-SSC Sp800-56Ar3	A3740 (for OCI) A3739 (for OCI + CyberStream)

KAS-ECC SP800-56Ar3	A5165 (for OCI)

2.4 Physical Scope

10 The physical boundary of the TOE includes the models shown in Table 5. The TOE is shipped to the customer via commercial courier.

Table 5: TOE models

Model	CPU
OCI (OS: Linux 3.10)	
NETSCOUT Omnis Cyber Intelligence Appliance	Intel® Xeon® Gold 6132 (Skylake)
CyberStream (OS: Linux 3.10)	
NETSCOUT 2795J	Intel® Xeon® Gold 6152 (Skylake)
NETSCOUT 9802J	
NETSCOUT 9807J	
NETSCOUT 9895J	

2.4.1 Guidance Documents

11 The TOE includes the following guidance documents (PDF):

a) Omnis Cyber Intelligence Server Administrator Guide, Software Version 6.3.4, Rev C

b) Omnis CyberStream Certified Appliance Administrator Guide, 6.3.4, Rev B

c) NETSCOUT Omnis Cyber Intelligence with CyberStream v6.3.4 Common Criteria Guide, v1.9

12 Registered users download the guidance documents from NETSCOUT's web portal. https://www.netscout.com/support-services

2.4.2 Non-TOE Components

13 The TOE operates with the following components in the environment:

a) Audit server

b) CA/OCSP Responder

2.4.3 Functions not included in the TOE Evaluation

14 Only those functions listed at 2.3 have been evaluated.

a) REST API

b) iDRAC and Supermicro BMC interfaces

3 Security Problem Definition

15 The Security Problem Definition is reproduced from the NDcPP.

3.1 Threats

Table 6: Threats

Identifier	Description
T.UNAUTHORIZED_ ADMINISTRATOR_ ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_ CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_ COMMUNICATION_ CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_ AUTHENTICATION_ ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_ COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and

Identifier	Description
	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

Table 7: Assumptions

Identifier	Description
A.PHYSICAL_ PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.
A.LIMITED_ FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality). In the case of vNDs, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of the distributed TOE run inside more than one virtual machine (VM) on a single VS. There are no other guest VMs on the physical platform providing non-Network Device functionality.

Identifier	Description
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 NDcPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_ UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_ CREDENTIALS_ SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.COMPONENTS_RU NNING	For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.
A.RESIDUAL_ INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

3.3 Organizational Security Policies

Table 8: Organizational Security Policies

Identifier	Description
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

4.1 Security Objectives for the TOE

Table 9: Security Objectives for the TOE

Identifier	Description
None	

4.2 Security Objectives for the Operational Environment

Table 10: Security Objectives for the Operational Environment

Identifier	Description
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.
OE.NO_THRU_ TRAFFIC_ PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

Identifier	Description
OE.ADMIN_ CREDENTIALS_ SECURE	The Administrator’s credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.COMPONENTS_RUNNING	For distributed TOEs, the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.
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.

5 Security Requirements

5.1 Conventions

16 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

e) **Iteration.** Indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

17 **Note:** Operations performed within the Security Target are denoted within brackets []. Operations shown without brackets are reproduced from the NDcPP.

5.2 Extended Components Definition

18 Table 11 identifies the extended components that are reproduced in this Security Target from the NDcPP.

Table 11: Extended Components

SFR	Source	TDs/RFI
FAU_GEN_EXT.1	NDcPP2.2e	N/A
FAU_STG_EXT.1	NDcPP2.2e	N/A
FAU_STG_EXT.4	NDcPP2.2e	N/A
FCO_CPC_EXT.1	NDcPP2.2e	N/A
FCS_HTTPS_EXT.1	NDcPP2.2e	N/A
FCS_RBG_EXT.1	NDcPP2.2e	N/A
FCS_SSHC_EXT.1	NDcPP2.2e	TD0636
FCS_SSHS_EXT.1	NDcPP2.2e	TD0631
FCS_TLSC_EXT.1	NDcPP2.2e	TD0634
FCS_TLSS_EXT.1	NDcPP2.2e	TD0555, TD0556, TD0569, TD0635
FIA_PMG_EXT.1	NDcPP2.2e	TD0571
FIA_UIA_EXT.1	NDcPP2.2e	N/A

SFR	Source	TDs/RFI
FIA_UAU_EXT.2	NDcPP2.2e	N/A
FIA_X509_EXT.1/Rev	NDcPP2.2e	TD0527
FIA_X509_EXT.1/ITT	NDcPP2.2e	TD0527
FIA_X509_EXT.2	NDcPP2.2e	TD0537
FIA_X509_EXT.3	NDcPP2.2e	N/A
FPT_SKP_EXT.1	NDcPP2.2e	TD0639
FPT_APW_EXT.1	NDcPP2.2e	N/A
FPT_TST_EXT.1	NDcPP2.2e	N/A
FPT_TUD_EXT.1	NDcPP2.2e	N/A
FPT_STM_EXT.1	NDcPP2.2e	TD0632
FTA_SSL_EXT.1	NDcPP2.2e	N/A

5.3 Functional Requirements

19 Table 12 provides a summary of the SFRs and identifies which distributed TOE component implements the SFR.

Table 12: Summary of SFRs

Requirement	Title	TOE Components
FAU_GEN.1	Audit Data Generation	All
FAU_GEN.2	User Identity Association	All
FAU_GEN_EXT.1	Security Audit Data Generation for Distributed TOE components	All
FAU_STG_EXT.1	Protected Audit Event Storage	All
FAU_STG_EXT.4	Protected Local Audit Event Storage for Distributed TOEs	All
FCO_CPC_EXT.1	Distributed TOE component registration channel.	All
FCS_CKM.1	Cryptographic Key Generation	All

Requirement	Title	TOE Components
FCS_CKM.2	Cryptographic Key Establishment	All
FCS_CKM.4	Cryptographic Key Destruction	All
FCS_COP.1/ DataEncryption	Cryptographic symmetric key operations	All
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)	All
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)	All
FCS_COP.1/ KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)	All
FCS_HTTPS_EXT.1	HTTPS Protocol	OCI
FCS_RBG_EXT.1	Random Bit Generation	All
FCS_SSHC_EXT.1	SSH Client Protocol	All
FCS_SSHS_EXT.1	SSH Server Protocol	All
FCS_TLSC_EXT.1	TLS Client Protocol without mutual authentication	OCI
FCS_TLSS_EXT.1	TLS Server Protocol	All
FIA_AFL.1	Authentication Failure Management	All
FIA_PMG_EXT.1	Password Management	All
FIA_UIA_EXT.1	Timing of user identification and provided services	All
FIA_UAU_EXT.2	Password-based Authentication Mechanism	All
FIA_UAU.7	Protected Authentication Feedback	All
FIA_X509_EXT.1/Rev	X.509 Certificate Validation	OCI
FIA_X509_EXT.1/ITT	X.509 Certificate Validation	OCI
FIA_X509_EXT.2	X.509 Certificate Authentication	OCI
FIA_X509_EXT.3	X.509 Certificate Requests	All
FMT_MOF.1/ ManualUpdate	Management of Security Functions Behaviour	All

Requirement	Title	TOE Components
FMT_MOF.1/Functions	Management of Security Functions Behaviour	All
FMT_MOF.1/Services	Management of Security Functions Behaviour	All
FMT_MTD.1/CoreData	Management of TSF Data	All
FMT_MTD.1/CryptoKeys	Management of TSF Data	All
FMT_SMF.1	Specification of Management Functions	All
FMT_SMR.2	Restrictions on Security Roles	All
FPT_ITT.1	Basic Internal TSF Data Transfer Protection	OCI is the client, CyberStream is the server
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)	All
FPT_APW_EXT.1	Protection of Administrator Passwords	OCI
FPT_TST_EXT.1	TSF Testing	All
FPT_TUD_EXT.1	Trusted Update	All
FPT_STM_EXT.1	Reliable Time Stamps	All
FTA_SSL_EXT.1	TSF-initiated Session Locking	All
FTA_SSL.3	TSF-initiated Termination	All
FTA_SSL.4	User-initiated Termination	All
FTA_TAB.1	Default TOE Access Banners	All
FTP_ITC.1	Inter-TSF trusted channel	All
FTP_TRP.1/Admin	Trusted Path	All

5.3.1 Security Audit (FAU)

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 shutdown of the audit functions;

b) All auditable events for the not specified level of audit; and

c) *All administrative actions comprising:*

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

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

Table 13: Audit Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FAU_STG_EXT.4	None.	None.
FCO_CPC_EXT.1	• Enabling communications between a pair of components. • Disabling communications between a pair of components.	Identities of the endpoint pairs enabled or disabled.
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.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
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.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time – either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if “lock session” is selected)	Any attempts at unlocking of an interactive session	None.

Requirement	Auditable Events	Additional Audit Record Contents
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failure of the trusted path functions.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure of certificate validation
	Any addition, replacement or removal of trust anchors in the TOE's trust store	Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.1/ITT	Unsuccessful attempt to validate a certificate	Reason for failure of certificate validation
	Any addition, replacement or removal of trust anchors in the TOE's trust store	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
FPT_ITT.1	Initiation of the trusted channel.	None

Requirement	Auditable Events	Additional Audit Record Contents
	Termination of the trusted channel.	None.
	Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_SSHC_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
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
FMT_MOF.1/Services	None	None
FMT_MTD.1/CryptoKeys	None	None
FMT_MOF.1/Functions	None	None.

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

a) Date and time of the event;

b) Type of event;

c) Subject and object identity (if applicable);

d) The outcome (success or failure) of the event;

e) *Additional information specified in column three of Table 13*

f) [no other information]

FAU_GEN_EXT.1 Security Audit Data Generation

FAU_GEN_EXT.1.1 The TSF shall be able to generate audit records for each TOE component. The audit records generated by the TSF of each TOE component shall include the subset of security relevant audit events which can occur on the TOE component.

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.

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 be a distributed TOE that stores audit data on the following TOE components: *OCI and CyberStream*

]

FAU_STG_EXT.1.3 The TSF shall [drop new audit data] when the local storage space for audit data is full.

FAU_STG_EXT.4 Protected Local Audit Event Storage for Distributed TOEs

FAU_STG_EXT.4.1 The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: [

OCI: [drop new audit data],

CyberStream: [drop new audit data]

].

5.3.2 Communication (FCO)

FCO_CPC_EXT.1 Component Registration Channel Definition

FCO_CPC_EXT.1.1 The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

FCO_CPC_EXT.1.2 The TSF shall implement a registration process in which components establish and use a communications channel that uses [

- A channel that meets the secure channel requirements in [FPT_ITT.1]]

for at least *TSF data*.

FCO_CPC_EXT.1.3 The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.

5.3.3 Cryptographic Support (FCS)

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: [

- 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
- FFC Schemes using ‘safe-prime’ groups that meet the following: “NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [RFC 3526].

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

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: [

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”
- FFC Schemes using “safe-prime” groups that meet the following: ‘NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [RFC 3526].

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

Application note: Modified by TD0580 and TD0581.

FCS_CKM.4 Cryptographic Key Destruction

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

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

] that meets the following: *No Standard*.

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

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

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 [

- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256, 384, 521 bits*]]

] that meet the following: [

- 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].

FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

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

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-256, HMAC-SHA-512] and cryptographic key sizes *[256, 512]* **and message digest sizes [256, 512] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”*.

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 HTTPS using TLS.

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

FCS_RBG_EXT.1 Random Bit Generation

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

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

FCS_SSHC_EXT.1 SSH Client Protocol

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [4344,5656,6668].

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

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

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

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521] as its public key algorithm(s) and rejects all other public key algorithms.

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

FCS_SSHC_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1] and [diffie-hellman-group14- sha256, diffie-hellman-group16-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521] are the only allowed key exchange methods used for the SSH protocol.

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

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

FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [4256, 4344, 5656, 6668].

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

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

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

Application Note: OCI supports CBC ciphers and CyberStream supports CTR ciphers.

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521] as its public key algorithm(s) and rejects all other public key algorithms.

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

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

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

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_ECDHE_ECDSA_WITH_AES128_GCM_SHA 256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES256_GCM_SHA 384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES128_CBC_SHA 256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES256_CBC_SHA 384 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] and no other curves] in the Client Hello.

FCS_TLSS_EXT.1 TLS Server 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_ECDHE_ECDSA_WITH_AES128_GCM_SHA 256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES256_GCM_SHA 384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES128_CBC_SHA 256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES256_CBC_SHA 384 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 [ECDHE curves [secp256r1] and no other curves]].

FCS_TLSS_EXT.1.4 The TSF shall support [no session resumption or session tickets, session resumption based on session tickets according to RFC 5077].

5.3.4 Identification and Authentication (FIA)

FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1-5*] 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 [*account is manually unlocked using the local console*] is taken by an Administrator; prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].

FIA_PMG_EXT.1 Password Management

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

g) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”];

h) Minimum password length shall be configurable to between [*5*] and [*255*] characters.

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.

FIA_UAU_EXT.2 Password-based Authentication Mechanism

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

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**.

FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates**.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 6960].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.

  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

FIA_X509_EXT.1/ITT X.509 Certificate Validation

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

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of two certificates.
- The certificate path must terminate with a trusted CA certificate.
- 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:
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

FIA_X509_EXT.1.2/ITT 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.

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 [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].

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.

5.3.5 Security Management (FMT)

FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

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

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*.

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*.

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*.

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*.

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 manage the cryptographic keys;*
    - *Ability to configure the cryptographic functionality;*
    - *Ability to configure the interaction between TOE components;*
    - *Ability to re-enable an Administrator account;*
    - *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;*
    - *Ability to manage the trusted public keys database].*

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.

5.3.6 Protection of the TSF (FPT)

FPT_APW_EXT.1 Protection of Administrator Passwords

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

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

FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure and **detect its** modification when it is transmitted between separate parts of the TOE **through the use of** [TLS, HTTPS].

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.

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].

FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [during initial start-up (on power on), periodically during normal operations] to demonstrate the correct operation of the TSF: [*software integrity, cryptographic module integrity, hardware integrity*].

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.

5.3.7 TOE Access (FTA)

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.

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*.

FTA_SSL.4 User-initiated Termination

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

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.

5.3.8 Trusted path/channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel (Refinement)

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

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [

- *Exporting audit]*

FTP_TRP.1 /Admin Trusted Path

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

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

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

5.4 Assurance Requirements

20 The TOE security assurance requirements are summarized in Table 14.

Table 14: Assurance Requirements

Assurance Class	Components	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 Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
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

21 In accordance with section 7.1 of the NDcPP, the following refinement is made to ASE:

a) ASE_TSS.1.1C Refinement: The TOE summary specification shall describe how the TOE meets each SFR. In the case of entropy analysis, the TSS is used in conjunction with required supplementary information on Entropy.

6 TOE Summary Specification

22 The following describes how the TOE fulfils each SFR included in section 18.

23 Refer to Table 12 for the summary of which distributed TOE component fulfils which SFR.

6.1 Security Audit

6.1.1 FAU_GEN.1 & FAU_GEN_EXT.1

24 The TOE generates the audit records specified at FAU_GEN.1 containing the following fields:

a) Date/Time

b) Type of event

c) Message (including user if applicable and indication of success or failure)

25 The following information is logged as a result of the Security Administrator generating/importing or deleting cryptographic keys:

a) Cryptographic key name

b) Storage location

c) Function performed.

26 Table 15 identifies the TOE components that generate the auditable events defined in FAU_GEN.1.1.

Table 15: Audit Events

Requirement	Auditable Events	TOE Component
FAU_GEN.1	Start-up and shutdown of the audit functions	All
	Administrative login and logout (Name of user account shall be logged if individual user accounts are required for Administrators)	All
	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)	All
	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)	All
	Resetting passwords (name of related user account shall be logged)	All

Requirement	Auditable Events	TOE Component
FCO_CPC_EXT.1	Enabling communications between a pair of components. Disabling communications between a pair of components. (Identities of the endpoints pairs enabled or disabled.)	All
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	OCI
FCS_SSHS_EXT.1	Failure to establish an SSH session	All
FCS_SSHC_EXT.1	Failure to establish an SSH session	All
FCS_TLSC_EXT.1	Failure to establish a TLS Session	OCI
FCS_TLSS_EXT.1	Failure to establish a TLS Session	All
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	All
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	All
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	All
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	OCI
FIA_X509_EXT.1/ITT	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	OCI
FIA_X509_EXT.2	X.509 Certificate Authentication	OCI
FIA_X509_EXT.3	X.509 Certificate Requests	All
FMT_MOF.1/ ManualUpdate	Any attempt to initiate a manual update	All
FMT_MOF.1/Functions	Modification of the behaviour of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full.	All

Requirement	Auditable Events	TOE Component
FMT_SMF.1	All management activities of TSF data.	All
FPT_ITT.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	All
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	All
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	All
FTA_SSL_EXT.1	The termination of a local session by the session locking mechanism.	All
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	All
FTA_SSL.4	The termination of an interactive session.	All
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	All
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	All

6.1.2 FAU_GEN.2

27 The TOE includes the user identity in audit events resulting from actions of identified users.

6.1.3 FAU_STG_EXT.1

28 The audit records are securely sent to a remote audit server in the operational environment using SSH. This prevents the audit records from unauthorized viewing and modification during transmission. Both TOE components transmit audit data to the remote audit server in real time using SSH.

29 The TOE logs all events related to startup/shutdown, external communications, user authentication, and user management (user creation/deletion, password changes, role changes) and administrative commands in the audit log.

30 The TOE is a distributed TOE with both components storing audit data locally. The local audit record will drop new audit data if it exceeds the storage capacity. The storage capacity to record audit data locally is 25 TB.

31 Only authorized administrators may view audit records and no capability to modify the audit records is provided.

6.1.4 FAU_STG_EXT.4

32 Each TOE component stores audit data locally. Each component when local audit storage is full, will drop new audit data.

6.2 Communication

6.2.1 FCO_CPC_EXT.1

33 Registration of the TOE's components is performed manually by the Security Administrator and uses a channel that meets the secure channel requirements in FPT_ITT.1.

34 To disable communication between the components, the Security Administrator can delete the CyberStream appliance entry using the Web GUI.

35 The minimum configuration is the deployment of an OCI and one Cyberstream. Multiple Cyberstreams can be deployed and maintain their own separate communication channels with the OCIs and external IT entities that comply with FPT_ITT.1 and FTP_ITC.1. The Cyberstreams do not communicate with each other.

6.3 Cryptographic Support

6.3.1 FCS_CKM.1

36 The TOE supports key generation for the following asymmetric schemes:

a) **ECC P-256, P-384, P-521.** Used in TLS and SSH.

b) **FCC Schemes using safe-prime:** Used in SSH.

6.3.2 FCS_CKM.2

37 The TOE cryptographic key mapping is listed in Table 16. The TOE supports the following key establishment schemes:

a) **NIST SP 800-56A conformant key establishment schemes:**

i) **EC Schemes.** Used in TLS and SSH. TOE is both sender and receiver.

ii) **FFC Safe Primes.** Used in SSH. TOE is both sender and receiver. The TOE meets NIST Special Publication 800-56A Revision 3 and RFC 3526.

Table 16: Cryptographic Key Mapping

Scheme	SFR	Service
EC Schemes	FCS_TLSS_EXT.1	GUI/Administration OCI & CyberStream TLS Server
	FCS_TLSC_EXT.1	OCI TLS Client

Scheme	SFR	Service
	FCS_SSHC_EXT.1	Audit Server
	FCS_SSHS_EXT.1	CLI / Administration
FFC Safe Primes	FCS_SSHS_EXT.1	CLI / Administration
	FCS_SSHC_EXT.1	Audit Server

6.3.3 FCS_CKM.4

38 Table 17 shows the origin, storage location and destruction details for cryptographic keys. Unless otherwise stated, the keys are generated by the TOE.

Table 17: Keys

Key	Algorithm	Storage	Zeroization
SSH Private Keys	ECDSA	Flash - plaintext	Stored in a protected file on the OS with root access only. Keys stored in non-volatile memory are destroyed by using the linux shred utility where the data is overwritten with pseudo-random numbers 15 times and a final overwrite of zeroes before removing the file
SSH Ephemeral Keys	AES / DH / ECDH	RAM – plaintext	Keys held in volatile memory are zeroized on de-allocation by finalizers
TLS Server Private Keys	ECDSA	Flash – plaintext	Stored in plaintext keystore. Keys are being zeroized by the Java 11 JSSE and Bouncy Castle (both being in FIPS mode). Zeroization occurs when the administrator invokes the Bouncy Castle keystore utility to specify the use of new keys. Zeroization of key material is managed via the JVM's garbage collection process.
TLS Client Ephemeral Keys	AES / ECDHE	RAM – plaintext	Bouncy Castle destroys with a single-pass overwrite of zeros. Keys held in volatile memory are zeroized on de-allocation by finalizers

6.3.4 FCS_COP.1/DataEncryption

39 All TOE components perform encryption and decryption using the AES algorithm with key sizes of 128 and 256 bits in CBC and GCM modes for TLS communication support. The implementation performs encryption and decryption using the AES algorithm with key sizes of 128 and 256 bits in CBC and CTR mode for SSH communication support.

40 Additionally, the AES algorithm with key size of 256 bits in CTR mode is implemented to support DRBG functionality (CTR_DRBG 256 bit).

41 The AES algorithm meets ISO 18033-3, CBC mode meets ISO 10116, CTR mode meets ISO 10116, and the GCM mode meets ISO 19772.

6.3.5 FCS_COP.1/SigGen

42 The TOE provides cryptographic signature generation and verification services using:

a) Elliptic Curve Digital Signature Algorithm (ECDSA) with 256, 384, and 521-bit key sizes using NIST curves of P-256, P-384 and P-521 for TLS communications in accordance with ISO/IEC 14888-3

6.3.6 FCS_COP.1/Hash

43 The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512.

44 SHA is implemented in the following parts of the TSF:

a) SHA-1 for Diffie-Hellman group 14 SSH key agreement scheme

b) SHA-256 for SSH HMAC message authentication support

c) SHA-1 and SHA-256 for TLS algorithm support (AES 128 and AES 256)

d) SHA-256, SHA-384, SHA-512 for ECDSA algorithm support (P-256, P384, P521)

e) SHA-256 for password hashing

6.3.7 FCS_COP.1/KeyedHash

45 The TOE provides keyed-hashing message authentication services using HMAC-SHA-256, HMAC-SHA-512

46 HMAC is implemented in the following protocols: SSH.

47 The characteristics of the HMACs used in the TOE are given in Table 18.

Table 18: HMAC Characteristics

Algorithm	Block Size	Key Size	Digest Size
HMAC-SHA-256	512 bits	256 bits	256 bits
HMAC-SHA-512	1024 bits	512 bits	512 bits

6.3.8 FCS_HTTPS_EXT.1

48 The TOE web GUI is accessed via an HTTPS connection using the TLS implementation described by FCS_TLSS_EXT.1. The TOE does not use HTTPS in a client capacity. The TOE’s HTTPS protocol complies with RFC 2818.

49 RFC 2818 specifies HTTP over TLS. The majority of RFC 2818 is spent on discussing practices for validating endpoint identities and how connections must be setup and torn down. The TOE web GUI operates on an explicit port designed to

natively speak TLS: it does not attempt STARTTLS or similar multi-protocol negotiation which is described in section 2.3 of RFC 2818.

6.3.9 FCS_RBG_EXT.1

50 All TOE components implement a NIST-approved counter deterministic random bit generator (CTR DRBG). All TOE components provide the same software-based entropy source as described in the proprietary entropy specification. The DRBG is seeded with a minimum of 256 bits of entropy so that it is sufficient to ensure full entropy for 256-bit keys, which are the largest keys generated by the TSF.

6.3.10 FCS_SSHC_EXT.1

51 The TOE's SSHv2 client implementation complies with RFCs 4251, 4252, 4253, 4254, 4344, 5656 and 6668. SSHv1 implementation is not supported. All TOE components initiate SSH communication for a secure connection to the operating environment's audit server for exporting audit records.

52 The SSHv2 Client only implements the following in the evaluated configuration:

a) authentication support: public key-based

b) encryption algorithm: AES128-CTR and AES256-CTR

c) public key for authentication: ecdsa-sha2-nistp521

d) host key for authentication: ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521

e) data integrity: hmac-sha2-256, hmac-sha2-512

f) key agreement scheme: diffie-Hellman-group14-sha1, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521

53 The SSH implementation will detect all large packets greater than 262,144 bytes and drop in accordance with RFC 4253. The SSH connections will be rekeyed after no more than one hour and no more than one gigabyte of transmitted data.

6.3.11 FCS_SSHS_EXT.1

54 The TOE's SSHv2 server implementation complies with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5656 and 6668. SSHv1 implementation is not supported. All TOE components act as a SSH server to secure connections for remote CLI administration.

55 The SSHv2 Server ONLY supports the following in the evaluated configuration:

a) authentication support: public key-based and password-based

b) encryption algorithm: AES128-CBC, AES256-CBC, AES128-CTR and AES256-CTR **NOTE:** OCI supports CBC ciphers and CyberStream supports CTR ciphers.

c) public key and host key for authentication: ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521

d) data integrity: hmac-sha2-256, hmac-sha2-512

e) key agreement scheme: Diffie-Hellman-group14-sha1

56 The SSH implementation will detect all large packets greater than 262,144 bytes and drop in accordance with RFC 4253. The SSH connections will be rekeyed after no more than one hour and no more than one gigabyte of transmitted data. Users are

verified when attempting to authenticate via public key through the use of the authorized_keys file.

6.3.12 FCS_TLSC_EXT.1

57 The TOE implements a TLS client for the trusted channel with its components.

58 TLS 1.2 is allowed and ciphersuites are restricted to the following:

a) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

b) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

c) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

d) TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

59 The TOE supports the use of DNS as reference identifiers. DNS is used in establishing communications between TOE components, channel registration and when performing upgrades to the TOE, and validates the CN field or SAN field. IP addresses are not supported.

60 The TOE will present Supported Elliptical Curve extensions in the Client Hello. The only allowable NIST curves are: secp256r1.

61 The TOE's TLS implementation will only support a wildcard in the left-most label (e.g. *.example.com). All other usages of a wildcard will cause a failure in the connection. The TOE does not support URI or service name reference identifiers or pinned certificates.

6.3.13 FCS_TLSS_EXT.1

62 TOE components operate as a TLS server.

63 The server only allows TLS protocol version 1.2 (rejecting any other protocol version) and is restricted to the following ciphersuites by default:

a) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

b) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

c) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

d) TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

64 Ciphersuites are not user-configurable.

65 The TLS server is capable of negotiating ciphersuites that include ECDHE key agreement schemes. The TOE supports key establishment using ecdhe curve secp256r1.

66 The OCI TLS server supports session resumption using session tickets according to RFC 5077. Session tickets are encrypted using AES-CBC symmetric algorithms, using key size of 128 consistent with FCS_COP.1/DataEncryption.

67 The CyberStream TLS server supports no session resumption or session tickets.

6.4 Identification and Authentication

6.4.1 FIA_AFL.1

68 The TOE is capable of tracking authentication failures of remote administrators logging in to the TOE using the (OCI) Web GUI or CLI.

69 When a user account has sequentially failed authentication the configured number of times (default 3), the account will be locked until the Security Administrator manually unlocks the account using the local console or a configurable time period has elapsed. When a user account is locked out it is only locked out remotely.

70 To set the maximum number of failed attempts and lock out time period for Web GUI user accounts, the administrator can use the designated menu in the GUI, or by editing /opt/NetScout/rtm/bin/serverprivate.properties file via CLI method.

71 The administrator can also configure the maximum number of failed attempts and lock-out time period for SSH connections by editing password-auth and system-auth files in the /etc/pam.d directory.

6.4.2 FIA_PMG_EXT.1

72 The TOE supports the local definition of administrative users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”.

73 The minimum password length is settable by the Administrator and can range from 5 to 255 characters.

6.4.3 FIA_UIA_EXT.1/FIA_UAU_EXT.2

74 The TOE requires all users to be successfully identified and authenticated. Each TOE component maintains its own local authentication mechanism. The only pre-authentication service allowed on the CLI is the display of the standard Linux pre-authentication banner; which can be configured by the Security Administrator through the CLI. The only pre-authentication service allowed at the Web GUI is the pre-authentication banner; which can be configured by the Security Administrator through the CLI.

75 Administrative access to the TOE is facilitated through one of several interfaces:

a) Directly connecting to the TOE appliance via a local serial connection

b) Remotely connecting to the TOE appliance via SSH

c) Remotely connecting to the TOE Web GUI via HTTPS

76 The Security Administrator available at the local and remote CLI is “backups”. The Security Administrator available for the Web GUI is “Administrator”. The OCI maintains a “backups” user and “Administrator” user. The CyberStream maintains a Security Administrator “backups” user. The “Administrator” and “backups” users are both Security Administrators with the same privileges and capable of executing administrative functions at the interfaces and devices they access.

77 The methods of authentication used for the local CLI versus the remote CLI can be configured separately even though they provide identical management functionality. The local CLI requires the user to authenticate to the TOE component’s local authentication mechanism with their username/password combination. The SSH CLI requires the user to authenticate with either a public key or username/password combination. The Web GUI requires the user to authenticate with a

username/password. The TOE then either grants administrative access if the correct credentials or public keys are provided or indicates that the login was unsuccessful.

6.4.4 FIA_UAU.7

78 For all authentication at the local CLI the TOE, no feedback or characters are displayed when the administrative password is entered so that the password is obscured.

6.4.5 FIA_X509_EXT.1/ITT/ FIA_X509_EXT.2/ FIA_X509_EXT.3/ FIA_X509_EXT.1/Rev

79 The TOE performs certificate validity checking for the TLS connection between TOE components. As part of the certificate validation checking, the OCI will validate certificate revocation status of Web GUI certificates, using an OCSP server in the Operational Environment. The OCI also performs revocation checks when validating certificates between TOE components. If the revocation status cannot be verified, the certificate will be rejected.

80 The TSF determines the validity of certificates by ensuring that the certificate and the certificate path are valid in accordance with RFC 5280. The OCI supports a minimum path length of three certificates for the Web GUI. All TOE components support a minimum path length of 2 certificates for inter-TOE certificates. In addition, the certificate path is terminated in a trusted CA certificate, the validity of dates are checked, the basicConstraints extension is present, and the CA flag is set to TRUE for all CA certificates. Finally, the TOE ensures the extendedKeyUsage field includes the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) for server certificates used in TLS, or the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) for OCSP certificates used for OCSP. There are no exceptions to the rules for extendedKeyUsage fields.

81 Each TOE component contains a trust store where root CA and intermediate CA certificates may be uploaded. The trust stores are not cached. If a certificate is deleted, it is immediately untrusted. If a certificate is added to the trust store, it is immediately trusted for its given scope. For the purposes of the web UI, a root CA and intermediate CA are required. For the purposes of the inter-TOE component communication, a root CA is required.

82 The OCI component chooses its server certificate by using the “nscertutil” to import the certificate, validate it against the CAs in the trust store and then placing it into the correct path. The CyberStream component chooses its server certificate by first importing the Certificate Response to the trust store, and then configuring the “is.pem” file with the certificates and key as a bundle.

83 In order to support HTTPS/TLS connectivity for the web UI interface and the inter-TOE component communication, the TSF of all components provide the ability to generate a Certificate Request Message as specified by RFC 2986 so that its server certificate can be signed by a Certification Authority. The message includes public key, Common Name, Organization, Organizational Unit, and Country values. The certificate chain of the Certificate Response is validated by the TSF prior to being installed as the TOE’s server certificate.

6.5 Security Management

6.5.1 FMT_MOF.1/ManualUpdate

84 The TOE restricts the ability to perform software updates to Security Administrators for all TOE components. Only manual updates are allowed in the evaluated configuration of the TOE.

6.5.2 FMT_MOF.1/Services

85 The TOE restricts the ability to start and stop services to Security Administrators for all TOE components.

6.5.3 FMT_MOF.1/Functions

86 The TOE restricts the ability to modify (enable/disable) transmission of audit records to an external audit server to Security Administrators for all TOE components.

6.5.4 FMT_MTD.1/CoreData

87 Users are required to login before being provided with access to any administrative functions. Access to TSF data and functions, including managing the TOE's trust store, is restricted to Security Administrators as described by FMT_SMR.2 below.

6.5.5 FMT_MTD.1/CryptoKeys

88 The TOE restricts the ability to manage SSH, TLS and any configured X.509 private keys to authenticated Security Administrators.

6.5.6 FMT_SMF.1

89 The TOE may be managed via the CLI (console & SSH) or GUI (HTTPS). The specific management capabilities include:

Table 19: TOE Component Management Capabilities

Management Capability	TOE Components	OCI Interfaces	CyberStream Interfaces
Ability to administer the TOE locally and remotely	All	CLI and GUI	CLI
Ability to configure the access banner (FTA_TAB.1)	All	CLI and GUI	CLI
Ability to configure the session inactivity time before session termination or locking (FTA_SSL_EXT.1, FTA_SSL.3)	All	CLI and GUI	CLI
Ability to update the TOE and to verify the updates (FMT_MTD.1/ManualUpdate, FPT_TUD_EXT.1)	OCI	CLI and GUI	N/A
Ability to configure the authentication failure parameters (FIA_AFL.1)	All	CLI and GUI	CLI

Management Capability	TOE Components	OCI Interfaces	CyberStream Interfaces
Ability to start and stop services	All	CLI	CLI
Ability to manage the cryptographic keys (FMT_MTD.1/CryptoKeys, FCS_CKM.1)	All	CLI	CLI
Ability to configure the cryptographic functionality (FCO_CPC_EXT.1)	CyberStream	N/A	CLI
Ability to set the time which is used for time-stamps	All	CLI	CLI
Ability to configure the interaction between TOE components (per FCO_CPC_EXT.1)	OCI	GUI	N/A
Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors	All	CLI	CLI
Ability to import X.509v3 certificates to the TOE's trust store	All	CLI	CLI
Ability to manage the trusted public keys database	All	CLI	CLI

6.5.7 FMT_SMR.2

90 The TOE operates 3 pre-defined users:

a) **backups.** Privileged access to the TOE via CLI. This profile equates to the Security Administrator role defined in this Security Target.

b) **Administrator.** Privileged access to the TOE via Web GUI. This profile equates to the Security Administrator role defined in this Security Target.

c) **unlocker.** Restricted access to the TOE via CLI. Limited to unlocking the backups user when the account has been locked due to authentication failures.

91 Management of TSF data via the CLI or web GUI is restricted to Security Administrators. To be considered the Security Administrator in the web GUI, the user must belong to the Security Administrator role.

6.6 Protection of the TSF

6.6.1 FPT_ITT.1

92 The TOE protects communication between the TOE components using HTTPS/TLS protocol and cipher suites.

6.6.2 FPT_APW_EXT.1

93 Administrator passwords are not stored by any TOE component in plaintext. All administrative passwords are hashed using SHA-256 and the hash is what is stored

by each TOE component. There is no function provided by the TOE to display a password value in plaintext. Each TOE component has its own local password store.

6.6.3 FPT_SKP_EXT.1

94 Symmetric or asymmetric keys cannot be viewed from TOE components. Security Administrators of the web UI or CLI are unable to view the keys stored in files or in memory. Table 17 shows the storage location of cryptographic keys.

6.6.4 FPT_TST_EXT.1

95 TOE components run the following tests on start-up:

a) POST or power on self-test: The POST memory test writes various data patterns into memory locations and reads them back to confirm that each memory location is functional. The test then interacts with every device in the machine looking for any failures. If any tests fail, the POST writes the failure indicator to the display and exits. When the POST ends successfully, the BIOS searches the various boot mechanisms (using the boot ordering maintained in ROM) for the operating system. The cryptographic POST consists of:

   i) software integrity test: HMAC-SHA1 verification of the binary code comprising the module executable

   ii) KATs (known answer tests) for cryptographic algorithms

   iii) PCTs (pairwise consistency tests) for asymmetric key pairs: a conditional test that runs only when asymmetric keys are generated

   iv) random bit and random number generator tests: (conditional tests that run only when random bits and random numbers are generated

96 All self-tests are performed by both TOE components at start-up.

97 In the event that a power on self-test fails, the boot process will terminate. The TOE component will need to be rebooted to attempt to clear the error. If the TOE component has been corrupted or the hardware has failed such that rebooting will not resolve the issue, a Security Administrator will need to contact NETSCOUT support. These tests and their response to failures is sufficient to ensure that the TSF behaves as described in the ST because it would detect any unauthorized modifications to the TOE, failures or tampering of the hardware (which could be an attempt to compromise its storage or take the TOE out of the range of operating conditions specified for its entropy source), and any cryptographic failures that could result in the establishment of insecure trusted channels.

6.6.5 FPT_TUD_EXT.1

98 Each TOE component is manually updated by the Security Administrator. The Security Administrator must download the updates along with the SHA256 hash associated with the file from the NETSCOUT support website to their local machine. The TOE component itself never communicates with the vendor support website.

99 The OCI is updated using the CLI and the CyberStream is updated via the OCI. The TOE is designed to function properly as individual components are updated. The current TOE version for each component can be queried through the use of the Web GUI.

6.6.6 FPT_STM_EXT.1

100 The TOE incorporates an internal clock for each TOE component, that is used to maintain date and time. The Security Administrator sets the date and time during initial TOE configuration and may change the time during operation manually.

101 The TOE makes use of time for the following:

a) Audit record timestamps

b) Session timeouts (lockout enforcement)

c) Certificate validation

6.7 TOE Access

6.7.1 FTA_SSL_EXT.1

102 The Security Administrator may configure the TOE to terminate an inactive local interactive session (CLI) following a specified period of time.

6.7.2 FTA_SSL.3

103 The Security Administrator may configure the TOE to terminate an inactive remote interactive session (CLI and Web UI) following a specified period of time. The Security Administrator may configure the TOE to terminate an active session after a specified period of time. The session timeout and inactive timeout values are indicated in minutes. Each TOE component enforces its own termination of an idle session and must be individually configured.

6.7.3 FTA_SSL.4

104 Administrative users may terminate their own sessions at any time using the ‘exit’ command. Administrative users may terminate their Web UI session at any time using the Logoff option in the Menu.

6.7.4 FTA_TAB.1

105 The TOE displays an administrator configurable message to users prior to login at the CLI and web GUI.

6.8 Trusted Path/Channels

6.8.1 FTP_ITC.1

The TOE provides the ability to secure sensitive data in transit to and from assured endpoints in the TOE’s Operational Environment. All TOE components are acting as a SSH client and are conformant to FCS_SSHC_EXT.1.

6.8.2 FTP_TRP.1/Admin

106 The TOE provides the following trusted paths for remote administration:

a) CLI over SSH per FCS_SSHS_EXT.1

b) Web GUI over HTTPS per FCS_HTTPS_EXT.1.1 (HTTPS uses TLS per FCS_TLSS_EXT.1)

7 Rationale

7.1 Conformance Claim Rationale

107 The following rationale is presented with regard to the PP conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is a distributed network device.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the NDcPP.

c) **Security objectives.** As shown in section 4, the security objectives are reproduced directly from the NDcPP.

d) **Security requirements.** As shown in section 5, the security requirements are drawn from the NDcPP. No additional requirements have been specified.

7.2 Security Objectives Rationale

108 All security objectives are drawn directly from the NDcPP.

7.3 Security Requirements Rationale

109 All security requirements are drawn directly from the NDcPP.

Table 20: SFR Rationale

Identifier	SFR Rationale
T.UNAUTHORIZED_ ADMINISTRATOR_ ACCESS	• The Administrator role is defined in FMT_SMR.2 and the relevant administration capabilities are defined in FMT_SMF.1 and FMT_MTD.1/CoreData, with optional additional capabilities in FMT_MOF.1/Services and FMT_MOF.1/Functions • The actions allowed before authentication of an Administrator are constrained by FIA_UIA_EXT.1, and include the advisory notice and consent warning message displayed according to FTA_TAB.1 • The requirement for the Administrator authentication process is described in FIA_UAU_EXT.2 • Locking of Administrator sessions is ensured by FTA_SSL_EXT.1 (for local sessions), FTA_SSL.3 (for remote sessions), and FTA_SSL.4 (for all interactive sessions) • The secure channel used for remote Administrator connections is specified in FTP_TRP.1/Admin • (Malicious actions carried out from an Administrator session are separately addressed by T.UNDETECTED_ACTIVITY) • (Protection of the Administrator credentials is separately addressed by T.PASSWORD_CRACKING).
T.WEAK_ CRYPTOGRAPHY	• Requirements for key generation and key distribution are set in FCS_CKM.1 and FCS_CKM.2 respectively

Identifier	SFR Rationale
	• Requirements for use of cryptographic schemes are set in FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash • Requirements for random bit generation to support key generation and secure protocols (see SFRs resulting from T.UNTRUSTED_COMMUNICATION_CHANNELS) are set in FCS_RBG_EXT.1 • Management of cryptographic functions is specified in FMT_SMF.1
T.UNTRUSTED_ COMMUNICATION_ CHANNELS	• The general use of secure protocols for identified communication channels is described at the top level in FTP_ITC.1 and FTP_TRP.1/Admin; for distributed TOEs the requirements for inter-component communications are addressed by the requirements in FPT_ITT.1 • Requirements for the use of secure communication protocols are set for all the allowed protocols in FCS_HTTPS_EXT.1, FCS_SSHC_EXT.1, FCS_SSHS_EXT.1, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2 • Optional and selection-based requirements for use of public key certificates to support secure protocols are defined in FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3
T.WEAK_ AUTHENTICATION_ ENDPOINTS	• The use of appropriate secure protocols to provide authentication of endpoints (as in the SFRs addressing T.UNTRUSTED_COMMUNICATION_CHANNELS) are ensured by the requirements in FTP_ITC.1 and FTP_TRP.1/Admin; for distributed TOEs the authentication requirements for endpoints in inter-component communications are addressed by the requirements in FPT_ITT.1 • Additional possible special cases of secure authentication during registration of distributed TOE components are addressed by FCO_CPC_EXT.1.
T.UPDATE_ COMPROMISE	• Requirements for protection of updates are set in FPT_TUD_EXT.1 • Additional optional use of certificate-based protection of signatures can be specified using FPT_TUD_EXT.2, supported by the X.509 certificate processing requirements in FIA_X509_EXT.1, FIA_X509_EXT.2 and FIA_X509_EXT.3 • Requirements for management of updates are defined in FMT_SMF.1 and (for manual updates) in FMT_MOF.1/ManualUpdate, with optional requirements for automatic updates in FMT_MOF.1/AutoUpdate
T.UNDETECTED_ ACTIVITY	• Requirements for basic auditing capabilities are specified in FAU_GEN.1 and FAU_GEN.2, with timestamps provided according to FPT_STM_EXT.1

Identifier	SFR Rationale
	• Requirements for secure transmission of local audit records to an external IT entity via a secure channel are specified in FAU_STG_EXT.1 • If (optionally) configuration of the audit functionality is provided by the TOE then this is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MOF.1/Functions.
T.SECURITY_ FUNCTIONALITY_ COMPROMISE	• Protection of secret/private keys against compromise is specified in FPT_SKP_EXT.1 • Secure destruction of keys is specified in FCS_CKM.4 • If (optionally) management of keys is provided by the TOE then this is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MTD.1/CryptoKeys • (Protection of passwords is separately covered under T.PASSWORD_CRACKING)
T.PASSWORD_ CRACKING	• Requirements for password lengths and available characters are set in FIA_PMG_EXT.1 • Protection of password entry by providing only obscured feedback is specified in FIA_UAU.7 • Actions on reaching a threshold number of consecutive password failures are specified in FIA_AFL.1 • Requirements for secure storage of passwords are set in FPT_APW_EXT.1.
T.SECURITY_ FUNCTIONALITY_ FAILURE	• Requirements for running self-test(s) are defined in FPT_TST_EXT.1