RedSeal Server v10.5

Security Target

Version 1.0

17 October 2025

Prepared for:

RedSeal, Inc.
1600 Technology Drive, 4th Floor
San Jose, CA 95110

Prepared by:

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, Maryland 21046

Revision History		
Date	**Author**	**Modifications**
10/09/2024	Leidos	Initial draft.
11/19/2024	Leidos	Incorporate review comments
11/27/2024	Leidos	Updates for CCTL review
01/29/2025	Leidos	Updates for check-in
05/07/2025	Leidos	Updates based on reviewer feedback and product hardware/software updates
09/29/2025	Leidos	Updates to interface claims and for Technical Decisions
10/17/2025	Leidos	Updates to address validator feedback

Contents

List of Tables

1 Security Target Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE identification, ST and TOE conformance claims, ST conventions, glossary and list of abbreviations.

The TOE is RedSeal Server v10.5 from RedSeal, Inc. The TOE is a network device providing a Network Infrastructure Security Management (NISM) platform able to identify attack risk and non-compliance in an enterprise network.

The focus of this evaluation is on the TOE functionality supporting the claims in the collaborative Protection Profile for Network Devices (NDcPP] and Functional Package for Secure Shell (SSH) [SSHPKG] – see Section 1.2 for specific version information). The security functionality specified in [NDcPP] and [SSHPKG] includes protection of communications between the TOE and external IT entities, identification and authentication of administrators, auditing of security-relevant events, ability to verify the source and integrity of updates to the TOE, and use of NIST-validated cryptographic mechanisms.

This ST includes the following additional sections:

- Product and TOE Description (Section 2)—provides an overview of the Product and the TOE and describes the physical and logical boundaries of the TOE
- Security Problem Definition (Section 3)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 4)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem
- IT Security Requirements (Section 5)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 6)—describes the security functions of the TOE and how they satisfy the SFRs
- Protection Profile Claims (Section 7)—provides rationale supporting the claims for conformance of the ST and the TOE to [NDcPP] and [SSHPKG]
- Rationale (Section 8)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1 Security Target, Target of Evaluation, and Common Criteria Identification

ST Title: RedSeal Server v10.5 Security Target

ST Version: Version 1.0

ST Date: 17 October 2025

TOE Identification: RedSeal Server v10.5

TOE Developer: RedSeal, Inc.

Evaluation Sponsor: RedSeal, Inc.

CC Identification: Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017.

The specific tested version was 10.5.2.

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Version 3.1 Revision 5, April 2017
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1 Revision 5, April 2017
  - Part 3 Conformant.

This ST and the TOE it describes are conformant to the following:

- *collaborative Protection Profile for Network Devices*, Version 3.0e, 06 December 2023 [NDcPP], including the following optional and selection-based SFRs: FAU_STG.1; FAU_STG_EXT.3; FCS_HTTPS_EXT.1; FCS_NTP_EXT.1; FCS_TLSC_EXT.1; FCS_TLSS_EXT.1; FIA_AFL.1; FIA_PMG_EXT.1; FIA_X509_EXT.1/Rev; FIA_X509_EXT.2; FIA_X509_EXT.3; FMT_MOF.1/Functions; FMT_MTD.1/CryptoKeys; and FPT_APW_EXT.1,
- Functional Package for Secure Shell (SSH), Version 1.0, 13 May 2021 [SSHPKG], including the following selection-based SFRs: FCS_SSHS_EXT.1.

The following NIAP Technical Decisions are applicable to the claimed Protection Profile/Package:

- TD0682 – Addressing Ambiguity in FCS_SSHS_EXT.1 Tests [SSHPKG]
  - This TD archives TD0666 and is applicable to the TOE but relates solely to evaluation activities so it does not affect the ST.
- TD0695– Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package [SSHPKG]
  - This TD modifies an explanation in the PP but does not change any requirements. This TD is applicable to the TOE.
- TD0732 – FCS_SSH_EXT.1.3 Inconsistency [SSHPKG]
  - This TD archives TD0694 and is applicable to the TOE but relates solely to the evaluation activities so it does not affect the ST.
- TD0777 – Clarification to Selections for Auditable Events for FCS_SSH_EXT.1 [SSHPKG]
  - This TD is applicable to the TOE.
- TD0836 – NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1 [NDcPP]
  - This TD is applicable to the TOE.
- TD0868 – NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8 [NDcPP]
  - This TD is not applicable because the TOE does not use IPSec
- TD0879 – NIT Technical Decision: Correction of Chapter Headings in CPP_ND_V3.0E [NDcPP]
  - This TD is not applicable because the TOE does not offer a local interface
- TD0880 – NIT Technical Decision: Removal of Duplicate Selection in FMT_SMF.1.1 [NDcPP]

  - This TD is an administrative update to the PP that removes a duplicate selection and changes the wording of the selection. This TD is applicable to the TOE.

- TD0886 – Clarification to FAU_STG_EXT.1 Test 6
  - This TD adds an application note to the test EA and is applicable to the TOE.
- TD0899 – NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2
  - This TD corrects a test activity in FCS_TLSC_EXT.1 and FCS_TLSS_EXT.1 that is applicable to the TOE.
- TD0900 – NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3
  - This TD updates FIA_UIA_EXT.1 and is applicable to the TOE.
- TD0909 – Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0
  - This TD is applicable to the TOE.
- TD0921 – NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment
  - This TD updates FCS_CKM.1 and is applicable to the TOE.
- TD0923 – NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2
  - This TD is applicable to the TOE but does not affect the contents of this ST as it only affects the application note.

1.3 Conventions

The following conventions are used in this document:

- Security Functional Requirements—Part 1 of the CC defines the approved set of operations that may be applied to functional requirements: iteration; selection; assignment; and refinement.
  - Iteration—allows a component to be used more than once with varying operations. In this ST, the only iterated requirements are those reproduced from [NDcPP], which uses descriptive strings to distinguish iterations of a requirement. For example, iterations of FCS_COP.1 are identified FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash.
  - Selection—allows the specification of one or more elements from a list. Selections completed in the ST are indicated using bold italics and are enclosed by brackets (e.g., [***selection***]).
  - Assignment—allows the specification of an identified parameter. Assignments completed in the ST are indicated using bold text and are enclosed by brackets (e.g., [**assignment**]). An assignment within a selection is identified in bold italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Refinement—allows the addition of details. Refinements made in the ST of requirements drawn from [NDcPP] would be indicated using bold for additions and strike-through for deletions (e.g., "… ~~some~~ **all** objects).
- Other sections of the ST—other sections of the ST use bolding and/or different fonts (such as `Courier`) to highlight text of special interest, such as captions, commands, or filenames specific to the TOE.

1.4 Abbreviations and Acronyms

Table 1: Abbreviations and Acronyms

Abbreviation	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CLI	Command Line Interface
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
HMAC	Hash-based Message Authentication Code
ICMP	Internet Control Message Protocol
IP	Internet Protocol
JDBC	Java Database Connectivity
MAC	Message Authentication Code
NTP	Network Time Protocol
PBKDF2	Password-Based Key Derivation Function—a key derivation function also used for password hashing.
PKCS	Public Key Cryptography Standards
RMI	Remote Method Invocation
SCP	Secure Copy
SFTP	SSH File Transfer Protocol
SHA	Secure Hash Algorithm
SMB	Server Message Block
SSH	Secure Shell
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
USB	Universal Serial Bus
VGA	Video Graphics Array

2 Product and TOE Description

2.1 Product Overview

The RedSeal Platform is a Network Infrastructure Security Management (NISM) platform that continuously identifies critical attack risk and non-compliance in complex enterprise security infrastructure. It provides organizations with an understanding of where security is working, where improvement is needed, and where the greatest cyber-attack risks lie.

RedSeal creates a model of the network based on information it collects from configuration files from switches, routers, firewalls and load balancers. RedSeal can integrate with public and private cloud managers to include all network environments in the network model. In addition, RedSeal imports host and vulnerability data from vulnerability scanners and other sources.

This network modeling is achieved without agents, span ports or taps and without being in line with production traffic or consuming net flow data.

The RedSeal Platform comprises the following components:

- RedSeal Server—the primary component of the RedSeal Platform, it controls data collection and analysis. It includes reporting and analytics engines and a threat reference library. It is the only component of the RedSeal Platform within the scope of evaluation.
- RedSeal Client (also identified as the RedSeal Java client, or just the Java client)—a Java Swing thick client that is installed on a user's workstation and can be used to manage a single instance of RedSeal Server. It is outside the TOE boundary, but is an optional component in the operational environment of the TOE.
- RedSeal Server Manager—can optionally be used to manage multiple RedSeal Servers, enabling centralized monitoring, administration, and management of one or more RedSeal Servers from a single administrative interface. It is intended to be used with 9 or more RedSeal Servers; provided and licensed separately; and excluded from the scope of this evaluation.

For this Security Target, the Target of Evaluation (TOE) is the RedSeal Server evaluated as a single network appliance device. The TOE claims exact conformance to the NDcPP and SSHPKG. As such, the security-relevant functionality of the product is limited to the claimed requirements in this PP and package. This product overview section (section 2.1) is intended to provide the reader with an overall summary of the entire product so that its intended usage is clear. The security-relevant functionality is described in sections 2.2 and 2.4.

2.2 TOE Overview

The TOE is RedSeal Server v10.5. It is a network device providing a Network Infrastructure Security Management (NISM) platform able to identify attack risk and non-compliance in an enterprise network.

The TOE requires users to be identified and authenticated before they can access any of the TOE functions. For each session, the user is required to log in prior to successfully establishing a session through which TOE functions can be exercised. The only capabilities allowed prior to users authenticating are the display of the warning banner before authentication, and the TOE may send Echo Reply in response to Echo Request ICMP messages received at the Management interface. The banner is displayed on every login attempt.

The TOE provides a Command Line Interface (CLI) for management and administration. The CLI is accessible remotely via SSH only (no local access is provided by the TOE). The TOE supports a single CLI user (**cliadmin**) that is equivalent to the Security Administrator role specified in [NDcPP].

The RedSeal Client (Java client) supports an administrator role (**uiadmin**) that can be used to create accounts for web interface and Java client users. It implements some functions that are local to the client (such as a local inactivity timeout and setting of a pre-login message) that are not provided by the TOE and are outside of the evaluated configuration. The Java client also provides access to a small subset of administrative capabilities that are implemented by the TOE, such as generating an X.509 certificate request and loading a CA certificate onto the TOE. Communications between the Java client and the TOE are protected by TLS, so although the Java client itself is not part of the TOE, it is allowed to be used for these tasks and is considered an authorized IT entity in the operational environment of the TOE. This Java client can be accessed either by launching a dedicated executable Java application on the remote management computer, or by launching the "remote client" on the TOE's web interface, which serves the same content over the same interface but in the web browser rather than the separate thick client. "Java client" interchangeably references both the standalone Java executable and the browser-based

Lastly, the TOE provides a browser-based interface referenced as the "Web Beta" client. The sole administrative function provided by this interface is to set the warning banner for users that access the product using this interface. This interface is used for the product's operational functionality that is outside the scope of the PP conformance claim. A deprecated "Legacy" client also exists but this is simply a reskin of the Web Beta client's end-user functionality and is outside the scope of the TOE because it does not support any management functions that apply to the PP conformance claim. Remote administrators access the Web Beta client via a web browser and communication is protected by HTTPS.

RedSeal makes updates to the TOE software available for download from its support web site. Software updates are provided as image files. The **cliadmin** uses the CLI to upload software images to the TOE and to query the version numbers of images (up to three) currently held on the TOE.

2.3 TOE Architecture

The TOE comprises the RedSeal Server application running on a Linux operating system, together with a database, all installed on a physical appliance provided by RedSeal. The product can be provisioned as a virtual appliance image and deployed on a virtual platform; however these virtual deployments are not being considered for evaluation.

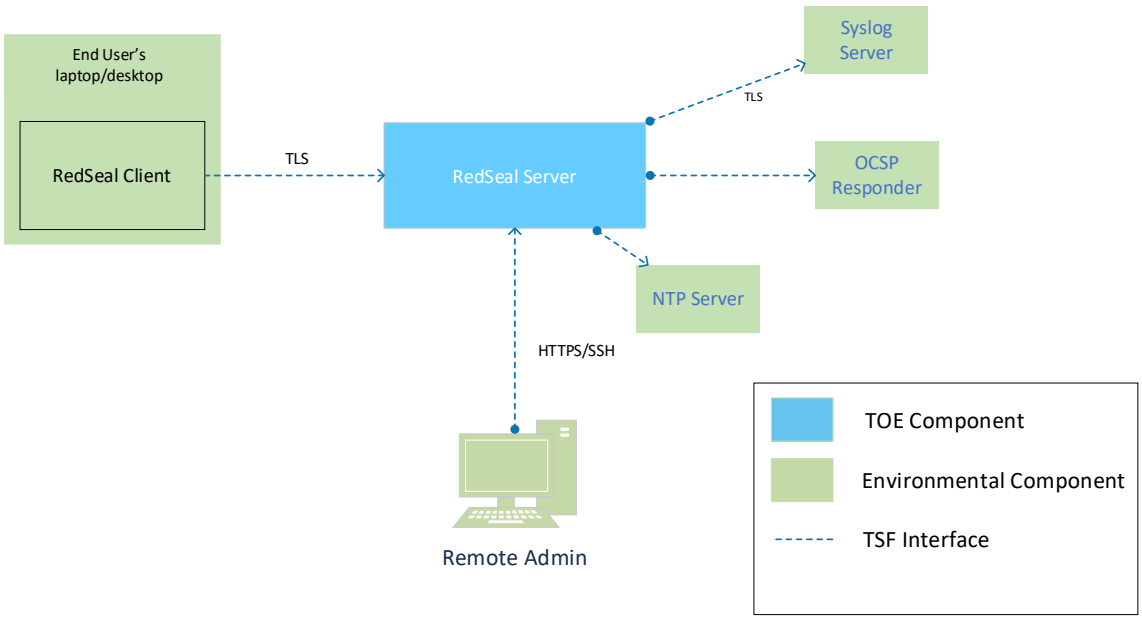
The RedSeal Server application includes the following main server processes:

- Admin server—provides administrative and infrastructure services to several other processes making up the RedSeal application
- RedSeal server—manages the import of network device configurations, contains the analysis engine, and provides the database interface.

The TOE includes a Linux distribution (RHEL 9.6) that provides the operating system on which the RedSeal application executes, and Temurin JRE 17.0.15+6, which supports part of the TOE's cryptographic algorithm implementation. The TOE (through an integrated Dell BSAFE SSL-J v7.3.1) implements cryptographic algorithms that support secure communication: between the TOE's browser-based Web Beta client and the remote administrative users (HTTPS); between the TOE and the Java client (TLS); between the TOE and the audit server (TLS); and between the TOE and legacy web clients (HTTPS). The TOE also includes OpenSSL 3.2.2-6, which provides the cryptographic algorithms to support SSH connections to the CLI.

Figure 1 shows the TOE in a sample deployment in its operational environment.

Figure 1 - TOE Boundary



2.4 Physical Boundaries

The TOE is available as a hardware appliance (denoted G5C). The following table summarizes the hardware appliance specifications.

Table 2: G5c Hardware Appliance Specifications

Height	1.7 in (43 mm)
Width	17.2 in (437 mm)
Depth	23.5 in (597 mm)
Weight	46 lbs (20 kg)
Temperature	50 – 95 degrees F (10 – 35 degrees C)
Humidity (noncondensing)	8 – 90 %
Voltage	100-240V, 8.5A-3.8A, 50-60 Hz
Processor	Intel Xeon Gold 5217 (Cascade Lake)
RAM	256 GB, 2933 MHz
Disk storage	Seagate 2.5", 1TB, SATA3 6Gb/s, 7.2K RPM, 512N, 128M
Power	Dual hot plug redundant (1 + 1) 700W

The TOE in its evaluated configuration requires the following components in its operational environment:

- Syslog server for external storage of exported audit records
- Administrative workstation equipped with SSH client software for remote administrator access to CLI
- Access to an OCSP Certification Authority for validating presented X.509 certificates.

The TOE in its evaluated configuration additionally supports the following optional components in its operational environment:

- NTP server
- Workstation with web browser for accessing the administrative Web Beta client
- Workstation running Windows or Mac OS X with Java 8 and web browser for downloading, installing and running the Java client.

Any function or interface not listed in this section is excluded from the scope of the TOE and is considered to be non-interfering with respect to the TSF.

2.5 Logical Boundaries

This section summarizes the following security functions provided by the TOE:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels.

2.5.1 Security Audit

The TOE generates audit records of security relevant events, including the events specified in [NDcPP] and [SSHPKG]. The TOE stores audit records locally and can also be configured to send the audit records to an external syslog server over a protected communication channel.

The logs comprising the audit trail are stored in the TOE's filesystem and protected from unauthorized modification and deletion by file system permissions. The TOE maintains a maximum of five log files—the current log file and four backups or archives. Each file has a default maximum of 50 megabytes (which is configurable by an administrator). When the current log file reaches its configured maximum size, it is closed and rotated to an archive, and a new current log file is created. If the maximum number of archive files already exists, the oldest one is deleted. The TOE will generate a warning message if the storage space for audit records reaches 75% capacity.

2.5.2 Cryptographic Support

The TOE implements cryptographic algorithms and mechanisms that provide random bit generation, asymmetric cryptographic key pair generation, key establishment, symmetric data encryption and decryption, digital signature generation and verification, cryptographic hashing, and keyed-hash message authentication services in support of higher level cryptographic protocols, including SSH and TLS.

2.5.3 Identification and Authentication

The TOE requires all users to be successfully identified and authenticated prior to accessing its security management functions and other capabilities. The TOE offers only remote access (via SSH) to a CLI (no local access); remote access (via HTTPS) to a browser-based administrative Web Beta client; and remote access (protected by TLS) using the Java client (either as a standalone Java application or the web-based Remote client) to support interactive administrator sessions.

The TOE provides a local password-based authentication mechanism for all users and enforces a minimum length for passwords. SSH public key authentication is also supported for the CLI. The TOE will deny remote access to a user after a configurable number of consecutive failed password authentication attempts (default is three).

2.5.4 Security Management

The TOE provides the security management functions necessary to configure and administer its security capabilities, including: configuring a login access banner; configuring a remote session inactivity time limit before session termination; configuring the parameters (number of consecutive failures, lockout period) for the authentication failure handling mechanism; setting the system date and time and also configuring NTP; performing software updates and verifying updates using a digital signature.

The TOE provides a CLI to access its security management functions. Administrators can access the CLI remotely using SSH (no local access provided). Additionally, some security management functions are accessible via the Web Beta client and the Java client. Security management commands are limited to administrators and are available only after they have been successfully identified and authenticated.

2.5.5 Protection of the TSF

The TOE protects sensitive data such as stored passwords and cryptographic keys so that they are not accessible even by an administrator.

The TOE provides reliable time stamps for its own use and can be configured to synchronize its time via NTP.

The TOE provides a trusted means for determining the current running version of its software and to update its software. The integrity of software updates can be verified using a digital signature.

The TOE implements various self-tests that execute during the power-on and start up sequence, including firmware/software integrity tests and cryptographic known answer tests that verify the correct operation of the TOE's cryptographic functions.

2.5.6 TOE Access

The TOE will terminate remote interactive sessions after a configurable period of inactivity. The TOE additionally provides the capability for administrators to terminate their own interactive sessions. The TOE can be configured to display an advisory and consent warning message before establishing a user session.

2.5.7 Trusted Path/Channels

The TOE protects interactive communication with remote administrators using SSH for remote access to the CLI (no local access to CLI provided); TLS using the Java client/Remote client (for remote GUI access to

the management interface whether through a standalone thick client or a browser-based Java implementation); and using HTTPS (for accessing the TOE's administrative Web Beta client).

The TOE is able to protect transmission of audit records to an external audit server using TLS.

2.6 TOE Documentation

The TOE is supplied with the following guidance documentation that describes the installation process for the TOE and provides guidance for configuration and secure use of its security features:

- RedSeal Installation and Administration Guide, Version 10.5.2
- RedSeal Server v10.5 Common Criteria Evaluated Configuration Guide (CCECG), Version 1.0

3 Security Problem Definition

This ST includes by reference the Security Problem Definition (comprising threat statements, assumptions, and organizational security policies) from [NDcPP]. The PP offers additional information about the threats, assumptions, and organizational security policies, but that has not been reproduced here and the PP should be consulted if there is interest in that material.

In general, the [NDcPP] has presented a Security Problem Definition appropriate for network infrastructure devices, and as such is applicable to the RedSeal Server. Note that the [NDcPP] defines several assumptions that only apply to the TOE in certain circumstances; within the context of this ST, the A.COMPONENTS_RUNNING assumption does not apply because the TOE is a standalone device, and the A.VS_TRUSTED_ADMINISTRATOR, A.VS_REGULAR_UPDATES, A.VS_ISOLATION, and A.VS_CORRECT_CONFIGURATION assumptions do not apply because the TOE does not have a virtual network device model.

4 Security Objectives

The [NDcPP] defines the following security objectives for the operational environment of the TOE.

Table 3: Security Objectives for the Operational Environment

Objective	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 TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

The [NDcPP] defines several objectives for the operational environment that only apply to the TOE in certain circumstances. Those that do not apply to the TOE (i.e., OE.COMPONENTS_RUNNING and OE.VM_CONFIGURATION) have not been included above. Additionally, the parts of the descriptions of the listed security objectives that only apply only to vNDs have not been included since the TOE does not have a virtual model (applies to OE.NO_GENERAL_PURPOSE, OE.TRUSTED_ADMIN, and OE.RESIDUAL_INFORMATION).

5 IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the TOE and to scope the evaluation effort.

The SFRs have all been drawn from [NDcPP] and [SSHPKG]. As such, operations on SFRs already performed in that PP and package are not identified here. Rather, the SFRs have been copied from [NDcPP] and [SSHPKG] and any formatting used in those documents have been removed. Operations performed on SFRs in the writing of this ST are identified in accordance with the conventions described in Section 1.3.

The SARs are the set of SARs specified in [NDcPP].

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from [NDcPP] and [SSHPKG]. The [NDcPP] or [SSHPKG] define the following extended SFRs and since they are not redefined in this ST, the [NDcPP] and [SSHPKG] should be consulted for more information in regard to these CC extensions.

- FAU_STG_EXT.1—Protected Audit Event Storage
- FAU_STG_EXT.3— Action in Case of Possible Audit Data Loss
- FCS_HTTPS_EXT.1—HTTPS Protocol
- FCS_NTP_EXT.1—NTP Protocol
- FCS_SSH_EXT.1 — SSH Protocol
- FCS_SSHS_EXT.1—SSH Server Protocol
- FCS_TLSC_EXT.1—TLS Client Protocol
- FCS_TLSS_EXT.1—TLS Server Protocol
- FCS_RBG_EXT.1—Random Bit Generation
- FIA_PMG_EXT.1—Password Management
- FIA_UIA_EXT.1—User Identification and Authentication
- FIA_X509_EXT.1—X.509 Certificate Validation
- FIA_X509_EXT.2—X.509 Certificate Authentication
- FIA_X509_EXT.3—X.509 Certificate Requests
- FPT_APW_EXT.1—Protection of Administrator Passwords
- FPT_SKP_EXT.1—Protection of TSF Data
- FPT_STM_EXT.1—Reliable Time Stamps
- FPT_TST_EXT.1—TSF Testing
- FPT_TUD_EXT.1—Trusted Update

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by RedSeal Server.

Table 4: TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1 Audit Data Generation
	FAU_GEN.2 User Identity Association
	FAU_STG.1 Protected Audit Trail Storage
	FAU_STG_EXT.1 Protected Audit Event Storage

Requirement Class	Requirement Component
	FAU_STG_EXT.3 Action in Case of Possible Audit Data Loss
FCS: Cryptographic support	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_NTP_EXT.1 NTP Protocol
	FCS_RBG_EXT.1 Random Bit Generation
	FCS_SSH_EXT.1 SSH Protocol
	FCS_SSHS_EXT.1 SSH Server Protocol
	FCS_TLSC_EXT.1 TLS Client Protocol
	FCS_TLSS_EXT.1 TLS Server Protocol
FIA: Identification and authentication	FIA_AFL.1 Authentication Failure Management
	FIA_PMG_EXT.1 Password Management
	FIA_UIA_EXT.1 User Identification and Authentication
	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: Security management	FMT_MOF.1/Functions Management of Security Functions Behavior
	FMT_MOF.1/ManualUpdate Management of Security Functions Behavior
	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: Protection of the TSF	FPT_APW_EXT.1 Protection of Administrator Passwords
	FPT_SKP_EXT.1 Protection of TSF Data (For Reading of All Symmetric Keys)
	FPT_STM_EXT.1 Reliable Time Stamps
	FPT_TST_EXT.1 TSF Testing
	FPT_TUD_EXT.1 Trusted Update
FTA: TOE Access	FTA_SSL.3 TSF-Initiated Termination
	FTA_SSL.4 User-Initiated Termination

Requirement Class	Requirement Component
	FTA_TAB.1 Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1 Inter-TSF Trusted Channel
	FTP_TRP.1/Admin Trusted Path

5.2.1 Security Audit (FAU)

5.2.1.1 Audit Data Generation (FAU_GEN.1)

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 administrator account shall be logged if individual 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 administrator account shall be logged)***].

d) Specifically defined auditable events listed in Table **5**.

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 **5**.

Table 5: Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
FAU_STG_EXT.3	Low storage space for audit events.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_NTP_EXT.1	Configuration of a new time server Removal of configured time server	Identity of new/removed time server
FCS_RBG_EXT.1	None.	None.
FCS_SSH_EXT.1	[***Failure to establish SSH connection***]	[***Reason for failure and Non-TOE endpoint of attempted connection (IP Address)***]
	[***Establishment of SSH connection***]	[***Non-TOE endpoint of connection (IP Address)***]
	[***Termination of SSH connection session***]	[***Non-TOE endpoint of connection (IP Address)***]
	[***Dropping of packet(s) outside defined size limits***]	[***Packet size***]
FCS_SSHS_EXT.1	No events specified	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 mechanisms.	Origin of the attempt (e.g., IP address).
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_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_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.

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

5.2.1.2 User Identity Association (FAU_GEN.2)

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 Protected Audit Trail Storage (FAU_STG.1)

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

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

5.2.1.4 Protected Audit Event Storage (FAU_STG_EXT.1)

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 maintain a [***log file***] of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4 The TSF shall be able to store [***persistent***] audit records locally with a minimum storage size of [**1,000 bytes per log file**].

FAU_STG_EXT.1.5 The TSF shall [***overwrite previous audit records according to the following rule: [the oldest archive audit file is deleted, the current audit file is rotated to be an archive audit file, and a new current audit file is created]***] when the local storage space for audit data is full.

FAU_STG_EXT.1.6 The TSF shall provide the following mechanisms for administrative access to locally stored audit records [***manual export***].

5.2.1.5 Action in Case of Possible Audit Data Loss (FAU_STG_EXT.3)

FAU_STG_EXT.3.1 The TSF shall generate a warning to inform the Administrator before the audit trail exceeds the local audit trail storage capacity.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 Cryptographic Key Generation (FCS_CKM.1)[1]

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, 3072-bit, 4096-bit] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;***
- ***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 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms", Section 6.6;***
- ***FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526, RFC 7919]***

].

5.2.2.2 Cryptographic Key Establishment (FCS_CKM.2)

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 meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";***
- ***FFC Schemes using "safe-prime" groups that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 3526, groups listed in RFC 7919].***

].

5.2.2.3 Cryptographic Key Destruction (FCS_CKM.4)

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

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

[1] Modified by TD0921

- 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 [a new value of the key]***;
  - ***instructs a part of the TSF to destroy the abstraction that represents the key***]

that meets the following: No Standard.

5.2.2.4 Cryptographic Operation (FCS_COP.1/DataEncryption)

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***].

5.2.2.5 Cryptographic Operation (FCS_COP.1/SigGen)[2]

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,***
- ***Elliptic Curve Digital Signature Algorithm***

]

and cryptographic key sizes [

- ***For RSA: [modulus 2048/3072/4096 bits],***
- ***For ECDSA: [256/384/521 bits]***

]

that meet the following: [

- ***For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 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 implementing [P-256, P-384, P-521] curves that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended curves"; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6***

].

[2] Modified by TD0921

5.2.2.6 Cryptographic Operation (FCS_COP.1/Hash)

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.

5.2.2.7 Cryptographic Operation (FCS_COP.1/KeyedHash)

FCS_COP.1.1/KeyedHash The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-1***, ***HMAC-SHA-256***, ***HMAC-SHA-384***, ***HMAC-SHA-512***] and cryptographic key sizes [**512 bits for HMAC-SHA-1 and HMAC-SHA-256, 1024 bits for HMAC-SHA-384 and HMAC-SHA-512**] and message digest sizes [***160***, ***256***, ***384***, ***512***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

5.2.2.8 HTTPS Protocol (FCS_HTTPS_EXT.1)

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.

5.2.2.9 NTP Protocol (FCS_NTP_EXT.1)

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

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

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

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

5.2.2.10 SSH Protocol (FCS_SSH_EXT.1)

FCS_SSH_EXT.1.1 The TSF shall implement SSH acting as a [***server***] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [***4344, 5656, 6668, 8268, 8308, 8332, 8731***] and no other standard.

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [

- ***"password" (RFC 4252),***
- ***"publickey" (RFC 4252): [***
  - ***ssh-rsa (RFC 4253)***

]

] and no other methods.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [**262105 bytes**] in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [

- ***aes128-ctr (RFC 4344),***

- ***aes256-ctr (RFC 4344)*,**
- ***aes128-cbc (RFC 4253)*,**
- ***aes256-cbc* (*RFC 4253)*,**

] and no other mechanisms.

FCS_SSH_EXT.1.5 The TSF shall protect data in transit from modification, deletion, and insertion using: [

- ***hmac-sha2-256 (RFC 6668),***
- ***hmac-sha2-512 (RFC 6668),***

] and no other mechanisms.

FCS_SSH_EXT.1.6 The TSF shall establish a shared secret with its peer using: [

- ***diffie-hellman-group14-sha256 (RFC 8268),***
- ***diffie-hellman-group16-sha512 (RFC 8268),***
- ***diffie-hellman-group18-sha512 (RFC 8268)***
- ***ecdh-sha2-nistp256 (RFC 5656),***
- ***ecdh-sha2-nistp384 (RFC 5656),***
- ***ecdh-sha2-nistp521 (RFC 5656)***

] and no other mechanisms.

FCS_SSH_EXT.1.7 The TSF shall use SSH KDF as defined in [

- ***RFC 4253 (Section 7.2),***
- ***RFC 5656 (Section 4)***

] to derive the following cryptographic keys from a shared secret: session keys.

FCS_SSH_EXT.1.8 The TSF shall ensure that: [

- ***a rekey of the session keys,***

] occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

5.2.2.11 SSH Server Protocol (FCS_SSHS_EXT.1)

FCS_SSHS_EXT.1.1 The TSF shall authenticate itself to its peer (SSH Client) using: [

- ***rsa-sha2-256 (RFC 8332),***
- ***rsa-sha2-512 (RFC 8332)***

].

5.2.2.12 TLS Client Protocol (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:

[

- ***TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***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***

] 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***].

FCS_TLSC_EXT.1.3 **T**he TSF shall not establish a trusted channel if the server certificate is invalid [

without any administrator override mechanism].

FCS_TLSC_EXT.1.4 The TSF shall [***present the Supported Groups Extension with the following curves/groups: [secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups***] in the Client Hello.

FCS_TLSC_EXT.1.5 The TSF shall [

- ***present the signature_algorithms extension with support for the following algorithms: [***
  - ***rsa_pkcs1 with sha256(0x0401),***
  - ***rsa_pkcs1 with sha384(0x0501),***
  - ***rsa_pkcs1 with sha512(0x0601),***
  - ***ecdsa_secp256r1 with sha256(0x0403),***
  - ***ecdsa_secp384r1 with sha384(0x0503),***
  - ***ecdsa_secp521r1 with sha512(0x0603),***
  - ***rsa_pss_rsae with sha256(0x0804),***
  - ***rsa_pss_rsae with sha384(0x0805),***
  - ***rsa_pss_rsae with sha512(0x0806),***
  - ***rsa_pss_pss with sha256(0x0809),***
  - ***rsa_pss_pss with sha384(0x080a),***
  - ***rsa_pss_pss with sha512(0x080b)***

  ] and no other algorithms;

].

FCS_TLSC_EXT.1.6 The TSF [***does not provide***] the ability to configure the list of supported ciphersuites as defined in FCS_TLSC_EXT.1.1.

FCS_TLSC_EXT.1.7 The TSF shall prohibit the use of the following extensions:

- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

FCS_TLSC_EXT.1.8 The TSF shall [***not use PSKs***].

FCS_TLSC_EXT.1.9 The TSF shall [***reject [TLS 1.2] renegotiation attempts***].

5.2.2.13 TLS Server Protocol (FCS_TLSS_EXT.1)

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_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***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***

] and no other ciphersuites.

FCS_TLSS_EXT.1.2 The TSF shall authenticate itself using X.509 certificate(s) using [

- ***RSA with key size [2048, 3072, 4096****] bits*].

FCS_TLSS_EXT.1.3 The TSF shall perform key exchange using [

- ***EC Diffie-Hellman key agreement over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves;***
- ***Diffie-Hellman parameters [ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192]***

].

FCS_TLSS_EXT.1.4 The TSF shall support [***no session resumption***].

FCS_TLSS_EXT.1.5 The TSF [***provides***] the ability to configure the list of supported ciphersuites as defined in FCS_TLSS_EXT.1.1.

FCS_TLSS_EXT.1.6 The TSF shall prohibit the use of the following extensions:

- Early data extension

FCS_TLSS_EXT.1.7 The TSF shall [***not use PSKs***].

FCS_TLSS_EXT.1.8 The TSF shall [***reject [TLS 1.2] renegotiation attempts***].

5.2.2.14 Random Bit Generation (FCS_RBG_EXT.1)

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 [SHA-256], CTR_DRBG (AES)***].

Application Note: *The SHA-256 Hash DRBG is used with BSAFE and CTR_DRBG (AES) is used with OpenSSL.*

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

5.2.3 Identification and Authentication (FIA)

5.2.3.1 Authentication Failure Management (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [**1-10**] 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 [action to unlock the Administrator account by entering the account unlock command] is taken by an Administrator; 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 Password Management (FIA_PMG_EXT.1)

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 [**7**] and [**15**] characters.

5.2.3.3 User Identification and Authentication (FIA_UIA_EXT.1)

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

- Display the warning banner in accordance with FTA_TAB.1;
- [***[respond to ICMP echo request packets, if enabled]***].

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_UIA_EXT.1.3[3] The TSF shall provide the following remote authentication mechanisms [***Web GUI password, Java client/Remote client password, SSH password, SSH public key***]. The TSF shall provide the following local authentication mechanisms [***none***].

Application Note: *The NDcPP is written in such a manner that it assumes the only GUI interface is a "web GUI", but there are no explicit requirements for remote administration that mandate this. The TOE also implements a GUI via a Java client/Remote client that uses TLS to interface with the TOE. The administrator authenticates to this GUI using a password. This is functionally identical to a "web GUI" as referenced by the SFR but is refined into the requirement as "Java Client/Remote client password" to signify that this is a GUI interface but not a web GUI specifically.*

In the context of this SFR, "Web GUI" refers to the Web Beta Client; the wording of "Web GUI" was kept to conform to the text of the selection.

FIA_UIA_EXT.1.4 The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

[3] Modified by TD0900

5.2.3.4 X.509 Certificate Validation (FIA_X509_EXT.1/Rev)

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

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of three certificates.
- The 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 DTLS/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 DTLS/TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

5.2.3.5 X.509 Certificate Authentication (FIA_X509_EXT.2)

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***].

5.2.3.6 X.509 Certificate Requests (FIA_X509_EXT.3)

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 Management of Security Functions Behavior (FMT_MOF.1/ManualUpdate)

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

5.2.4.2 Management of Security Functions Behavior (FMT_MOF.1/Functions)

FMT_MOF.1.1/Functions The TSF shall restrict the ability to [***determine the behaviour of***, ***modify the behaviour of***] the functions [***handling of audit data***] to Security Administrators.

5.2.4.3 Management of TSF Data (FMT_MTD.1/CoreData)

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

5.2.4.4 Management of TSF Data (FMT_MTD.1/CryptoKeys)

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

5.2.4.5 Specification of Management Functions (FMT_SMF.1)

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

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;

- [
  - ***Ability to configure local audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full; changes to local audit storage size)***;[4]
  - ***Ability to modify the behaviour of the transmission of audit data to an external IT entity***;
  - ***Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1***;
  - ***Ability to manage the cryptographic keys;***
  - ***Ability to configure the list of supported (D)TLS ciphers;***
  - ***Ability to re-enable an Administrator account***;
  - ***Ability to set the time which is used for time-stamps***;
  - ***Ability to configure NTP***;
  - ***Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors***;

[4] Modified by TD0880

- ***Ability to generate Certificate Signing Request (CSR) and process CA certificate response;***
- ***Ability to configure the authentication failure parameters for FIA_AFL.1;***
- ***Ability to manage the trusted public keys database;***

].

5.2.4.6 Restrictions on Security Roles (FMT_SMR.2)

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 remotely

are satisfied.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 Protection of TSF Data (for reading of all symmetric keys) (FPT_SKP_EXT.1)

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

5.2.5.2 Protection of Administrator Passwords (FPT_APW_EXT.1)

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.

5.2.5.3 Reliable Time Stamps (FPT_STM_EXT.1)

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

FPT_STM_EXT.1.2 The TSF shall [***allow the Security Administrator to set the time, synchronise time with an NTP server***].

5.2.5.1 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1[5] The TSF shall run a suite of the following self-tests

- *During initial start-up (on power on) to verify the integrity of the TOE firmware and software;*
- *Prior to providing any cryptographic service and [**at no other time**] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;*
- *[**start-up] self-tests [Cryptographic known answer tests, Pairwise consistency tests**].*

to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2 The TSF shall respond to [***all failures***] by [***entering a maintenance mode***].

[5] Modified by TD0836

5.2.5.2 Trusted Update (FPT_TUD_EXT.1)

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 [***the most recently installed version of the TOE firmware/software***].

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

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

5.2.6 TOE Access (FTA)

5.2.6.1 TSF-Initiated Termination (FTA_SSL.3)

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

5.2.6.2 User-Initiated Termination (FTA_SSL.4)

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

5.2.6.3 Default TOE Access Banners (FTA_TAB.1)

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 Inter-TSF Trusted Channel (FTP_ITC.1)

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

FTP_ITC.1.2 The TSF shall permit [***the TSF***] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [***export of audit records to external syslog server***].

5.2.7.2 Trusted Path (FTP_TRP.1/Admin)

FTP_TRP.1.1/Admin The TSF shall be capable of using [***SSH, 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 Security Assurance Requirements

The mandatory security assurance requirements for the TOE are included by reference from the [NDcPP]. None of the optional ALC_FLR.1, ALC_FLR.2 or ALC_FLR.3 security assurance requirements are claimed.

Table 6: Assurance Components

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1 Basic functional specification
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.1 Labelling of the TOE
	ALC_CMS.1 TOE CM coverage
ASE: 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
ATE: Tests	ATE_IND.1 Independent testing – conformance
AVA: Vulnerability assessment	AVA_VAN.1 Vulnerability survey

6 TOE Summary Specification

This section describes the following security functions implemented by the TOE to satisfy the SFRs claimed in Section 5.2:

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

6.1 Security Audit

6.1.1 Audit Data Generation

The TOE generates audit records of the following events:

- Start-up and shut-down of the TOE appliance—since the audit function is always enabled, this is equivalent to auditing start-up and shut-down of the audit function
- Administrative login and logout
- All management activities of TSF data, including changes to TSF data related to configuration changes and what was changed
- Generating/import of, changing, or deleting of cryptographic keys—the TOE logs the specific administrator action that was performed and identifies cryptographic keys either by identifying the certificate associated with the key, using the certificate subject identifier and certificate issuer identifier.
- Resetting passwords, including identification of the relevant user account

Additionally, the TOE logs the specifically defined auditable events listed in Table 5.

Each audit record generated by the TOE includes the following information, at a minimum: date and time stamp of when the auditable event occurred; type of event; identity of the subject that initiated the auditable event; the outcome of the event (e.g., success or failure); and 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 5. Audit records of events resulting from the actions of identified users include the relevant user identity.

This aspect of the Security Audit security function satisfies FAU_GEN.1 and FAU_GEN.2.

6.1.2 Audit Storage and Audit Record Export

The TOE is a single standalone appliance that is able to store generated audit records locally (i.e., on the appliance). The TOE maintains the following logs that together constitute the audit trail:

- Audit—contains records of all configuration changes made in the CLI
- System—contains records of system events, including server starts, stops, and restorations
- Server—contains all log messages generated by TOE server and database processes, including records in the Audit, System, and Analyzer log files.

For each of these logs, the TOE maintains a configurable number of log files on the appliance—the current log file and the most recently created log files up to the configured number minus one. For example, if the TOE is configured to maintain five files (the default) for each log, the log will consist of the current file and the four most recently created log files.

Log files are rotated based on a configurable schedule that can be specified in terms of size. The **cliadmin** can configure logs to be rotated when they reach a maximum size. The size must be in the range of 1,000 Bytes to 1,000MB (default is 50MB). When the current log file (for each of the logs defined above) reaches its rotation threshold, it is closed and a new current log file is created. If the configured maximum number of files for that log already exists, the oldest one is deleted. The maximum amount of space available on the TOE for all logs is 2GB. The TOE will generate a warning message if the storage space for audit records reaches 75% capacity. The warning is logged to the audit trail.

The logs comprising the audit trail are stored in the TOE's file system and protected from unauthorized modification and deletion by file system permissions. Log files are deleted when the oldest log is rotated out by the log rotation settings.

The TOE can be configured to export audit records to an external audit server over a trusted channel protected by TLS. In this circumstance, the TOE acts as a TLS client. The audit records are exported in real time (i.e., as they are generated). Additionally, the TOE provides a manual export mechanism that allows Administrators to download the locally stored audit records in order to view them.

This aspect of the Security Audit security function satisfies FAU_STG.1, FAU_STG_EXT.1, and FAU_STG_EXT.3.

6.2 Cryptographic Support

The TOE incorporates two cryptographic modules, used by the TOE as follows:

- OpenSSL—the TOE includes OpenSSL 3.2.2-6 to support SSH connections to the CLI (SSH server functionality).
- RSA BSAFE—the TOE includes an integrated Dell BSAFE SSL-J 7.3.1, which it uses to support secure communications with external IT for the Java client (TLS server), and web user access (HTTPS/TLS server). The TOE also uses BSAFE to support secure communications with remote users accessing the browser-based Web Beta client (HTTPS/TLS Server).

6.2.1 Cryptographic Operations

The TOE includes NIST-validated cryptographic algorithms providing supporting cryptographic functions. The following functions implemented by the OpenSSL cryptomodule included in the TOE have been certified in accordance with the identified standards.

OpenSSL provides the cryptographic algorithms to support SSH connections to the CLI (SSH Server).

Table 7: Cryptographic Functions Implemented by OpenSSL

Functions	Standards	Certificates
Asymmetric Key Generation (FCS_CKM.1)		
RSA (2048 bits)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.1	A7460 RSA KeyGen (FIPS 186-5)

Functions	Standards	Certificates
ECDSA (P-256, P-384, P-521 curves)	FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Appendix A.2	A7460 ECDSA KeyGen/KeyVer (FIPS 186-5)
FFC Schemes using ‘safe-prime’ groups: Diffie-Hellman MODP Groups 14, 16, 18	NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and RFC 3526	CCTL tested against known-good implementation
Key establishment (FCS_CKM.2)		
Elliptic curve-based scheme (P-256, P-384, P-521 curves)	NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”	A7460 KAS-ECC-SSC SP800-56Ar3
FFC Schemes using “safe-prime” groups: Diffie-Hellman MODP Groups 14, 16, 18	NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and groups listed in RFC 3526	A7460 KAS-FFC-SSC SP800-56Ar3
Data encryption (FCS_COP.1/DataEncryption)		
AES in CBC mode (128, 256 bits) AES in CTR mode (128, 256 bits)	ISO 18033-3 (AES) ISO 10116 (CBC and CTR mode)	A7460 AES-CBC A7460 AES-CTR
Digital signature generation and verification (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (2048 bit modulus)	FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Section 5.4, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	A7460 RSA SigGen/SigVer (FIPS 186-5)
Cryptographic hashing (FCS_COP.1/Hash)		
SHA-1 (digest size 160 bits) SHA-256 (digest size 256 bits) SHA-384 (digest size 384 bits) SHA-512 (digest size 512 bits)	ISO/IEC 10118-3:2004	A7460 SHA-1 A7460 SHA2-256 A7460 SHA2-384 A7460 SHA2-512
Keyed-hash message authentication (FCS_COP.1/KeyedHash)		
HMAC-SHA-1 (key size 512 bits, digest size 160 bits) HMAC-SHA-256 (key size 512 bits, digest size 256 bits) HMAC-SHA-512 (key size 1024 bits, digest size 512 bits)	ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”	A7460 HMAC-SHA-1 A7460 HMAC-SHA2-256 A7460 HMAC-SHA2-512
Deterministic random bit generation (FCS_RBG_EXT.1)		
CTR_DRBG (AES-256)	ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”	A7460 Counter DRBG

The following functions implemented by the TOE (via RSA BSAFE SSL-J) have been certified in accordance with the identified standards.

Table 8: Cryptographic Functions Implemented by RSA BSAFE SSL-J

Functions	Standards	Certificates
Asymmetric Key Generation (FCS_CKM.1)		
RSA (2048, 3072, 4096 bits)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.1	A7459 RSA KeyGen (FIPS 186-5)
ECDSA (P-256, P-384, P-521 curves)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2	A7459 ECDSA KeyGen/ KeyVer (FIPS 186-5)
FFC Schemes using 'safe-prime' groups: ffdhe2048, ffdhe3072; ffdhe4096; ffdhe6144; and ffdhe8192	NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and groups listed in RFC 7919	CCTL tested against known-good implementation
Key establishment (FCS_CKM.2)		
Elliptic curve-based scheme (P-256, P-384, P-521 curves)	NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	A7459 KAS-ECC-SSC SP800-56Ar3
FFC Schemes using 'safe-prime' groups: ffdhe2048, ffdhe3072; ffdhe4096; ffdhe6144; and ffdhe8192	NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and groups listed in RFC 7919	A7459 KAS-FFC-SSC SP800-56Ar3
Data encryption (FCS_COP.1/DataEncryption)		
AES in CBC mode (128, 256 bits) AES in GCM mode (128, 256 bits) AES in CTR mode (128, 256 bits)	ISO 18033-3 (AES) ISO 10116 (CBC and CTR mode) ISO 19772 (GCM mode)	A7459 AES-CBC A7459 AES-GCM A7459 AES-CTR
Digital signature generation and verification (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (2048, 3072, 4096 bit modulus)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS2v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	A7459 RSA SigGen/SigVer (FIPS 186-5)
ECDSA Digital Signature Algorithm (NIST curves P-256, P-384, P-521)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves"; ISO/IEC 14888-3, Section 6.4	A7459 ECDSA SigGen/SigVer (FIPS 186-5)

Functions	Standards	Certificates
Cryptographic hashing (FCS_COP.1/Hash)		
SHA-1 (digest size 160 bits) SHA-256 (digest size 256 bits) SHA-384 (digest size 384 bits) SHA-512 (digest size 512 bits)	ISO/IEC 10118-3:2004	A7459 SHA-1 A7459 SHA2-256 A7459 SHA2-384 A7459 SHA2-512
Keyed-hash message authentication (FCS_COP.1/KeyedHash)		
HMAC-SHA-1 (key size 512 bits, digest size 160 bits) HMAC-SHA-256 (key size 512 bits, digest size 256 bits) HMAC-SHA-384 (key size 1024 bits, digest size 384 bits) HMAC-SHA-512 (key size 1024 bits, digest size 512 bits)	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"	A7459 HMAC-SHA-1 A7459 HMAC-SHA2-256 A7459 HMAC-SHA2-384 A7459 HMAC-SHA2-512
Deterministic random bit generation (FCS_RBG_EXT.1)		
Hash_DRBG(SHA-256)	ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions"	A7459 Hash DRBG

The TOE performs AES encryption and decryption in accordance with ISO 18033-3, with key sizes of 128 and 256 bits, in the following modes of operation: CBC mode, as specified in ISO 10116; CTR mode as specified in ISO 10116; and GCM mode as specified in ISO 19772.

The TOE provides cryptographic signature services using the RSA Digital Signature Algorithm with a key size (modulus) of 2048 bits, in accordance with FIPS 186-5, "Digital Signature Standard (DSS)" for both TLS and SSH. For TLS, the TOE also supports 3072, 4096 bit modulus key sizes, and ECDSA Digital Signature Algorithm using NIST curves P-256, P-384, and P-521.

The TOE performs cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512, in accordance with ISO/IEC 10118-3:2004. The TOE uses the SHA hash algorithms as follows:

- as part of the HMAC algorithms that provide data integrity for SSH and TLS
- as part of RSA digital signature generation and verification
- as part of the conditioning used to protect stored passwords (refer to Section 6.5.1)
- for NTP authentication (SHA1).

The TOE performs keyed-hash message authentication in accordance with ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2". The following table summarizes the hash function, key length, block size, and output MAC lengths used by the HMAC function.

Table 9: HMAC Function Values

Hash Function	Key Length	Block Size	Output MAC Length
SHA-1	512 bits	512 bits	160 bits
SHA-256	512 bits	512 bits	256 bits
SHA-384	1024 bits	1024 bits	384 bits

SHA-512	1024 bits	1024 bits	512 bits

This aspect of the Cryptographic Support security function satisfies FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash.

6.2.2 Random Bit Generation

The TOE instantiates the SHA-256 Hash DRBG provided by its BSAFE module to generate random bits for use with other BSAFE algorithms and the AES-256 Counter DRBG provided by OpenSSL to generate random bits for use with other OpenSSL algorithms.

The TOE seeds these DRBGs with 256 bits minimum assumed entropy obtained from the hardware-based Intel RDRAND function. The TOE uses Java Native Instructions (JNI) to interface with the native C code used to interface with RDRAND to obtain seed data for the BSAFE cryptographic module.

This aspect of the Cryptographic Support security function satisfies FCS_RBG_EXT.1.

6.2.3 Cryptographic Key Generation and Establishment

The TOE generates RSA asymmetric key pairs with cryptographic key sizes (modulus) of 2048 bits, 3072, bits, and 4096 bits, in accordance with Appendix A.1 of FIPS PUB 186-5, “Digital Signature Standard (DSS)”. The RSA keys are used in support of SSH public key authentication and TLS server authentication. All three key sizes are used by Dell BSAFE SSL-J in support of TLS server authentication; 2048-bit key sizes are used by OpenSSL in support of SSH public key authentication.

The TOE generates ECC asymmetric key pairs over NIST curves P-256, P-384, and P-521, in accordance with Appendix A.2 of FIPS PUB 186-5, “Digital Signature Standard (DSS)”. The ECDSA keys are used in support of SSH and TLS key exchange.

The TOE generates FFC asymmetric safe-prime key pairs using Diffie-Hellman MODP groups 14, 16, and 18 in accordance with RFC 3526, Section 3. These keys are used in support of SSH key exchange.

The TOE generates FFC asymmetric safe-prime key pairs using FFDHE groups: ffdhe2048, ffdhe3072; ffdhe4096; ffdhe6144; and ffdhe8192 in accordance with RFC7919. These keys are used in support of TLS key exchange.

The TOE acts as both a sender and recipient in the SSH key establishment schemes. It acts as a server for the SSH management interface.

The TOE acts as both the client and the server for TLS key establishment schemes. It acts as a client for export of audit records to an external audit server, and as a server for remote administration via the RedSeal client and the web interfaces.

The following table summarizes the key establishment schemes implemented by the TOE, the relevant protocol SFR for each scheme, and the TOE services associated with each scheme.

Table 10: Key Establishment Scheme Usage by TOE

Scheme	SFR	Service
ECDH	FCS_TLSC_EXT.1	Audit server
ECDH	FCS_TLSS_EXT.1	Administration

Scheme	SFR	Service
DH (Groups ffdhe2048, ffdhe3072; ffdhe4096; ffdhe6144; and ffdhe8192)	FCS_TLSC_EXT.1	Audit server
DH (Groups ffdhe2048, ffdhe3072; ffdhe4096; ffdhe6144; and ffdhe8192)	FCS_TLSS_EXT.1	Administration
ECDH	FCS_SSHS_EXT.1	Administration
DH (MODP Group 14, 16, 18)	FCS_SSHS_EXT.1	Administration

This aspect of the Cryptographic Support security function satisfies FCS_CKM.1 and FCS_CKM.2.

6.2.4 Cryptographic Key Destruction

The TOE uses the following secret keys, private keys, and critical security parameters (CSPs).

Table 11: Private Keys, Symmetric Keys, and CSPs

Key/CSP	Origin, Use and Storage
RSA private key	Generated by TOE. Used to authenticate the TOE in TLS and SSH sessions. Stored on disk in PKCS #12 keystore.
EC DH private key	Generated by TOE. Used in TLS and SSH key exchange. Stored in RAM.
DH private key	Generated by TOE. Used in TLS and SSH key exchange. Stored in RAM.
AES keys used for secure communication	Generated by TOE. Used to encrypt/decrypt data transmitted/received in TLS and SSH sessions. Stored in RAM.
AES key used to encrypt PKCS #12 keystore	Derived by TOE from password using PBKDF2. Exists ephemerally in RAM.
HMAC keys	Generated by TOE. Used to verify integrity of packets in TLS and SSH sessions. Stored in RAM.
NTP keys	Specified by administrator. Used to authenticate communications received from configured NTP servers. Stored in plaintext in TOE file system.
DRBG parameters (seed, entropy input)	Generated by TOE. Used to instantiate DRBG. Stored in RAM.

The keys in the above table that are stored in RAM are ephemeral keys that are destroyed by the cryptomodule manipulating the key. The OpenSSL cryptomodule destroys keys directly by overwriting them once with zeroes. The BSAFE cryptomodule provides the `<object>.clearSensitiveData()` method, which destroys the reference to the key and immediately issues a request for garbage collection to destroy keys it holds in RAM.

The keys in the above table stored in plaintext in the TOE's file system are long-term persistent keys that are required for correct continuing operation of the product. They are destroyed when no longer required in one of two ways:

- The key is overwritten by a new value for the key, e.g., when the **cliadmin** changes the value of an NTP key using the `set ntp authentication symmetric add-key` CLI command
- The **cliadmin** executes the `reset all` CLI command, which resets the appliance to its factory defaults. The `reset all` command instructs a part of the TSF to destroy the abstraction that represents the key, using the Linux `rm –rf <filename>` command.

This aspect of the Cryptographic Support security function satisfies FCS_CKM.4.

6.2.5 Cryptographic Protocols

The TOE implements the following cryptographic protocols to protect communications between itself and non-TOE entities:

- TLS as a client—the TOE acts as a TLS client when exporting audit records to an external audit server
- TLS as a server—the TOE acts as a TLS server supporting inbound communications from the Java client
- TLS as a server—the TOE acts as a TLS server supporting inbound communications for remote user access via browser-based Remote/Web Beta clients
- HTTPS—in conjunction with TLS, the TOE supports the use of HTTPS for remote administrative access via the browser-based Remote/Web Beta clients
- SSH-2 as a server—the TOE acts as an SSH server supporting inbound remote administration via the CLI
- NTP—the TOE can synchronize its system clock with an NTP server.

6.2.5.1 TLS Client Protocol

In its evaluated configuration (after the **cliadmin** has executed the `enable common criteria` CLI command), the TOE supports TLS v1.2 with the following TLS cipher suites when acting as a TLS client:

- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- 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

The TOE presents the Supported Elliptic Curves Extension in its Client Hello message, specifying the following curves/groups: secp256r1; secp384r1; secp521r1; ffdhe2048, ffdhe3072; ffdhe4096; ffdhe6144; and ffdhe8192. This is done by default and is not configurable.

The TOE presents the signature_algorithms extension with support for the following algorithms by default (this is not configurable):

- rsa_pkcs1 with sha256(0x0401)
- rsa_pkcs1with sha384(0x0501)
- rsa_pkcs1 with sha512(0x0601)
- ecdsa_secp256r1 with sha256(0x0403)
- ecdsa_secp384r1 with sha384(0x0503)
- ecdsa_secp521r1 with sha512(0x0603)
- rsa_pss_rsae with sha256(0x0804)
- rsa_pss_rsae with sha384(0x0805)
- rsa_pss_rsae with sha512(0x0806)
- rsa_pss_pss with sha256(0x0809)
- rsa_pss_pss with sha384(0x080a)
- rsa_pss_pss with sha512(0x080b)

The TOE establishes the reference identifier for the external audit server based on the hostname (fully qualified domain name – FQDN; IP addresses are not supported) of the audit server configured by the **cliadmin** using the `set log` CLI command. The TOE verifies that the presented identifier matches the reference identifier according to RFC 6125 section 6, and establishes a trusted channel only if the server certificate is valid. The TOE verifies the external server's presented identifier by comparing it to the configured reference identifier, matching the server's FQDN. The TOE does not support wildcards for peer authentication. Certificate pinning is not supported.

The TOE's TLS client does not use PSKs and prohibits the use of the following extensions:

- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

The TOE rejects all renegotiation attempts for its TLS interfaces.

In addition to the `enable common criteria` CLI command, the TOE provides the ability to configure the list of supported ciphersuites for the TOE's TLS client using the command: `<enable|disable> ciphersuites.`

This aspect of the Cryptographic Support security function satisfies FCS_TLSC_EXT.1.

6.2.5.2 TLS Server Protocol

The TOE acts as a TLS server supporting communications with the Web Beta and Java clients. In its evaluated configuration (after the **cliadmin** has executed the `enable common criteria` CLI command), the TOE supports TLS v1.2 and the following TLS cipher suites when acting as a TLS server:

- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- 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

In its evaluated configuration, the TOE will accept ClientHello messages that specify support for TLS v1.2 and will reject ClientHello messages that specify support for other versions of TLS/SSL.

The TOE authenticates itself using an X.509 certificate with 2048, 3072, or 4096 bit RSA signature.

When the TOE negotiates a cipher suite that uses DHE as its key exchange algorithm, it sends a Server Key Exchange message that specifies the supported Diffie-Hellman parameters (ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192*).*

When the TOE negotiates a cipher suite that uses ECDHE as its key exchange algorithm, it sends a Server Key Exchange message that specifies the supported NIST curves (secp256r1, secp384r1, and secp521r1), the ECDH public key, and the associated domain parameters.

The TOE's TLS server does not use PSKs and prohibits the use of the Early data extension.

The TOE rejects all renegotiation attempts for its TLS interfaces and does not support session resumption.

In addition to the `enable common criteria` CLI command, the TOE provides the ability to configure the list of supported ciphersuites for the TOE's TLS client using: `<enable|disable cipher-suites>`.

This aspect of the Cryptographic Support security function satisfies FCS_TLSS_EXT.1.

6.2.5.3 HTTPS

The TOE uses HTTPS to secure communications with remote users accessing the TOE. The TOE implements a TLS server to permit the inbound remote administration traffic (HTTPS) and inbound non-administrative traffic (HTTPS) in which the peer initiates handshake and peer authentication is performed either via username and password or SSH-key based credentials. The TOE's HTTPS protocol complies with RFC 2818 and is implemented using TLS 1.2 (RFC 5246). HTTPS is used for the browser-based web connections, therefore, only the server requirements in RFC 2818 are applicable. RFC 2818 (section 2.1: complies as specified, section 2.2: complies as specified, section 2.2.1: not applicable, section 2.2.2: complies as specified, section 2.3: default port is 443, section 2.4: Use 'https' as specified, section 3.1: not applicable, and section 3.2: client identity checking is not performed/not applicable).The TOE uses HTTPS to protect communications between itself and remote users accessing the browser-based web interfaces. The TOE implements the server side of the HTTPS protocol according to RFC 2818 by using a TLS session to secure the HTTP connection. All HTTP data is sent as TLS "application data". In the event the TOE is presented with a peer certificate, the TOE will not establish the connection if the peer certificate is deemed invalid.

This aspect of the Cryptographic Support security function satisfies FCS_HTTPS_EXT.1.

6.2.5.4 SSH

The TOE acts as an SSH server when supporting inbound remote administration via the CLI. The TOE's implementation of SSH-2 complies with RFCs 4251, 4252, 4253, 4254, 4344, 5656, 6668, 8268, 8332, and 8731. It supports both public key and password-based authentication as described in RFC 4252.

During initial configuration of the TOE, the **cliadmin** uses the enable common criteria CLI command to disable disallowed algorithms that are otherwise supported by the TOE.

In the TOE's evaluated configuration, the SSH server implementation supports only the following algorithms and methods, and rejects all others:

- Encryption algorithms—aes128-cbc, aes256-cbc, aes128-ctr, and aes256-ctr
- User public key authentication algorithm—ssh-rsa
- Host key signature algorithm—rsa-sha2-256, rsa-sha2-512
- Data integrity MAC algorithms—hmac-sha2-256 and hmac-sha2-512
- Key exchange methods
    - diffie-hellman-group14-sha256 (RFC 8268),
    - diffie-hellman-group16-sha512 (RFC 8268),
    - diffie-hellman-group18-sha512 (RFC 8268)
    - ecdh-sha2-nistp256 (RFC 5656),
    - ecdh-sha2-nistp384 (RFC 5656),
    - ecdh-sha2-nistp521 (RFC 5656)
- KDF as defined in RFC 4253 (Section 7.2) and RFC 5656 (Section 4) to derive session keys from a shared secret

The TOE ensures that packets greater than 1 gigabyte in an SSH transport connection are dropped—i.e., such a packet is not processed further when this size limit is reached and the buffer containing the packet is freed. The TOE ensures 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 (transmitted or received). The TOE initiates a rekey when any of the thresholds is reached, whichever is hit first.

This aspect of the Cryptographic Support security function satisfies FCS_SSH_EXT.1 and FCS_SSHS_EXT.1.

6.2.5.5 NTP Protocol

The TOE can synchronize its system clock with an NTP server. The TOE supports NTP v4 as defined in RFC 5905 and uses SHA-1 as its means for authenticating the NTP timestamps it receives from configured NTP servers. The TOE can support up to five NTP time sources and will not update NTP timestamps from broadcast or multicast addresses. The **cliadmin** uses the `set ntp` CLI command to configure the NTP servers to be used by the TOE and the `set ntp authentication` CLI command to enable NTP server authentication and to configure the key to be used.

This aspect of the Cryptographic Support security function satisfies FCS_NTP_EXT.1.

6.3 Identification and Authentication

6.3.1 User Identification and Authentication

The TOE offers no services to external entities prior to identification and authentication, other than to display the advisory notice and consent warning message prior to completing the establishment of an interactive user session, and to respond to ICMP echo request (ping) packets, if enabled to do so. The TOE requires each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

The TOE defines two default administrator roles:

- **cliadmin**—performs administrative tasks on the CLI, remotely via SSH. This is the only user account able to access the CLI (i.e., it is not possible to create any new CLI users)
- **uiadmin**—performs administrative tasks remotely using the Web Beta client (HTTPS) and the Java client/Remote client (TLS).

In order to log in, the user must provide an identity and also authentication data that matches that identity configured on the TOE. Users logging on to **cliadmin** remotely via SSH can be configured to authenticate with either a password-based mechanism or a public key. At least one **cliadmin** must be configured to login using SSH public key in order to ensure continuity of administrator access in the event that all administrators using a password become locked-out due to the lockout function.

The TOE supports the following logon methods:

- Remote connection to the CLI via SSH v2, using username and password—the user uses SSH client software on their local workstation to initiate a connection, providing the **cliadmin** username and submitting the associated password when prompted. If the logon is successful, the user is presented with the CLI command prompt. The user terminates the remote connection to the CLI by entering the `exit` command at the CLI prompt.
- Remote connection to the CLI via SSH v2, using username and public key—the user uses SSH client software on their local workstation to initiate a connection, providing the **cliadmin** username. The client software additionally submits evidence (a digital signature of information in the authentication request, generated with the user's private key) that the user possesses the private key associated with the public key configured for the user on the TOE. Assuming the user's public key is associated with the **cliadmin** username on the TOE, and the TOE can verify the digital signature included in the authentication request, the logon is successful and the user is presented with the CLI command prompt. The user terminates the remote connection to the CLI by entering

the `exit` command at the CLI prompt. In the evaluated configuration at least one **cliadmin** must be configured to login using SSH public key in order to ensure continuity of administrator access in the event that all administrators using a password become locked-out due to the lockout function.

- Remote connection via the Java client, using username and password—the user launches the Java application on their local workstation and provides the following information:
  - Host Address—IP address or host name of the RedSeal server to connect to
  - Port—access port for the RedSeal server (3825 by default)
  - Username—RedSeal user account ID
  - Password—password for the entered RedSeal account
- Remote connection to the TOE's Remote client (i.e. the browser-based Java implementation of the Java client) , using username and password—the user navigates to the /rc/ui subdirectory of the TOE's web server root in a web browser. The user is presented with the same login prompt as with the Java client.
- Remote connection to the TOE's Web Beta client, using username and password—the user navigates to the /redseal/a/login subdirectory of the TOE's web server root in a web browser. The user is presented with a login prompt to enter their username and password.

This aspect of the Identification and Authentication security function satisfies FIA_UIA_EXT.1.

6.3.2 Authentication Failure Management

The TOE implements a mechanism to respond to consecutive failures to authenticate a remote login attempt using a password, within a specified time window (failure window). This mechanism is disabled by default. The **cliadmin** enables it by executing the `enable common criteria` CLI command. By default, the TOE allows three attempts to enter a valid password within a 15-minute failure window. If the configured number of failed attempts is reached within the failure window, the TOE locks the user account and prevents the user from logging in. The mechanism applies to all remote password-based interfaces, including the browser-based Web Beta client, the Java client, and the CLI. The default lockout period is 600 seconds (10 minutes). In the evaluated configuration the **cliadmin** must be configured to login using SSH public key in order to ensure continuity of administrator access in the event that all other administrators using a password become locked-out due to the lockout function. In the event that a user becomes locked out due to the consecutive authentication failure lockout function, the **cliadmin** can use the `enable user` command to unlock the user or the **uiadmin** can use the Java Client or Web Beta client to unlock the user: `EDIT -> System Settings -> Users` -> Right click user and click Edit. Alternatively, the user can wait for the configured lockout period to pass.

The **cliadmin** is able to individually configure the failed attempts threshold, the lockout duration, and the failure window using the `set property server` CLI command to set the following server properties appropriately:

- Failed attempts—`redseal.srm.authentication.max_failure_count`. The number of failed attempts can be set to a number in the range 1 to 10.
- Lockout period—`redseal.srm.authentication.lockout_duration_seconds`. The lockout period can be set to zero meaning the account is locked until such time as an administrator unlocks the account. Where the lockout period is configured for a non-zero number of seconds

(from 1 to a maximum of any integer), an administrator (**cliadmin** or **uiadmin**) is still able to unlock the account prior to the expiration of the lockout period.

- Failure window—`redseal.srm.authentication_failure_window_duration_seconds`.

Note, if the number of consecutive failed login attempts does not reach the configured failed attempts value within the configured failure window, the count of failed attempts is reset to zero. Therefore, the administrator is advised to set the failure window to a large enough value that a potential attacker is slowed down just as much as if there was no failure window and an account was locked for a period of time after the configured number of failure attempts was reached.

This aspect of the Identification and Authentication security function satisfies FIA_AFL.1.

6.3.3 Password Management

The TOE provides capabilities to manage passwords for the **cliadmin** and **uiadmin** accounts from the CLI. Management of passwords for user accounts created for the Web Beta client, Java client, is performed using the Web Beta client or Java client.

Passwords for the **cliadmin** and **uiadmin** accounts can be composed of any combination of upper and lower case letters, numbers, and the following special characters: `!@#$%^&*()_-+={}|[]\:";'<>?,./`.

The TOE implements two password policies, controlled by the `redseal.srm.strictPasswordCheck` server property. When this property is set to `false` (the default), the TOE enforces a minimum password length of seven characters. When the property is set to `true`, the TOE enforces a minimum password length of 15 characters. The property is also set to true when the **cliadmin** executes the `enable common criteria` CLI command.

This aspect of the Identification and Authentication security function satisfies FIA_PMG_EXT.1.

6.3.4 X.509 Certificate Validation

The TOE performs RFC 5280 certificate validation and certificate path validation on all X.509 certificates presented to it for the purpose of TLS server authentication. It also performs validation on the chain of certificates that are loaded into the TOE's trust store. The TOE supports a path length of at least three certificates.

The TOE validates a certification path by ensuring the presence of the basicConstraints extension with the CA flag set to TRUE for all CA certificates. The TOE will not treat a certificate as a CA certificate if the basicConstraints extension is not present or the CA flag is not set to TRUE. The certification path terminates with a trusted CA certificate designated as the Root CA.

The TOE validates X.509 certificates using the path validation algorithm defined in RFC 5280, which can be summarized as follows:

- The public key algorithm and parameters are checked
- The current date/time is checked against the validity period of the certificate
- The revocation status is checked
- The issuer name is checked to ensure that it equals the subject name of the previous certificate in the path

- Name constraints are checked, to make sure the subject name is within the permitted subtrees list of all previous CA certificates and not within the excluded subtrees list of any previous CA certificate
- The asserted certificate policy OIDs are checked against the permissible OIDs of the previous certificate, including any policy mapping equivalencies asserted by the previous certificate
- Policy constraints and basic constraints are checked, to ensure that any explicit policy requirements are not violated and that the certificate is a CA certificate, respectively
- The path length is checked to ensure that it does not exceed any maximum path length asserted in this or a previous certificate
- The key usage extension is checked
- Any other critical extensions are recognized and processed.

The certificate chain is validated to the root, and a revocation check is performed on each certificate (except the root certificate) using OCSP.

The TOE uses the following rules for validating the extendedKeyUsage[6] 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.
- 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.

The TOE downloads and caches OCSP status information for every CA listed in the trusted CA list of the TOE. The OCSP status is cached for the 'next update time' that is configured on the OCSP responder. The TOE uses this received value as the cache time. OCSP responders can also be configured for other external devices if someone decides to use it. The TOE uses a hard coded 1 hour as next update time (cached time) in this case. Caching only applies to validated certificates; if a TOE never validated a certificate, the TOE cache does not store the OCSP information for the issuing CA. To use OCSP for verifying the revocation status of certificates, you must configure the TOE to access an OCSP responder (server). The entity that manages the OCSP responder can be a third-party certificate authority (CA) or, if your enterprise has its own PKI, the TOE itself.

This aspect of the Identification and Authentication security function satisfies FIA_X509_EXT.1/Rev.

6.3.5 X.509 Certificate Authentication

The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication of the external TLS syslog server. The TOE uses the X.509 certificate presented in the TLS server Certificate message to authenticate the TLS server, using the validation algorithm described above. Root CA certificates and intermediate CA certificates that may be needed as part of the validation are stored in the TOE's trust store. The administrative guidance provides instructions for uploading root and intermediate CA certificates to the TOE's trust store.

[6] Certificates are not used for trusted updates or executable code integrity, nor does the TOE support TLS mutual authentication. Therefore, the TOE does not support the rules for validating certificates with the Code Signing or Client Authentication purpose in the extendedKeyUsage field, and this part of the requirement is trivially satisfied.

As described above, the TOE uses OCSP to determine the revocation status of X.509 certificates. If the TOE is unable to establish a connection to the OCSP responder in order to determine the validity of a certificate, it will drop the connection.

This aspect of the Identification and Authentication security function satisfies FIA_X509_EXT.2.

6.3.6 X.509 Certificate Requests

The **cliadmin** is able to use the `create cert-request` CLI command to generate a Certificate Request as specified in RFC 2986. In order to specify the values of the Common Name (CN), Organization, Organizational Unit, and Country to be included in the certificate request, the **cliadmin** first generates a self-signed certificate using the `create self-certificate` CLI command. This command prompts the **cliadmin** for specific attributes, including CN, Organization, Organizational Unit, and Country. These values will then be used in the certificate request generated by the `create cert-request` command.

The **cliadmin** uses the `upload certificate` CLI command to load the certificate returned by the CA in response to the certificate request. In order for this command to succeed, the **cliadmin** must first use the `upload ca-certificate` CLI command to upload the chain of certificates from the Root CA into the TOE's trust store. The TOE will then validate the chain of certificates from the Root CA when the **cliadmin** uploads the certificate returned by the CA.

Alternatively, the **uiadmin** (or a Java client user with Admin permissions) can use the **Server Certificate** tab of the Java client **System Settings** dialog to generate a Certificate Request and to upload the certificate returned by the CA, as well as the chain of certificates from the Root CA.

This aspect of the Identification and Authentication security function satisfies FIA_X509_EXT.3.

6.4 Security Management

6.4.1 Security Roles

The TOE implements the following two default administrator roles that together provide the capabilities of the Security Administrator role:

- **cliadmin**—performs administrative tasks using the CLI, remotely via SSH
- **uiadmin**—performs administrative tasks, remotely, using the Java client/Remote client (TLS) or Web Beta client (via HTTPS/TLS).

The cliadmin has full permissions, while the uiadmin and users with admin permissions have a subset of administrative permissions (see section 6.4.2 below).

Security management commands provided by the CLI, Java client/Remote client, and the Web Beta client are limited to administrators and are available only after they have provided acceptable user identification and authentication data to the TOE. For the CLI, the **cliadmin** is the only user able to access the CLI, so its security management functions are restricted to the **cliadmin**. For the Java client/Remote client and Web Beta client, access to security management functions is restricted to **uiadmin**.

This aspect of the Security Management security function satisfies FMT_SMR.2.

6.4.2 Specification of Management Functions

The TOE provides the following security management functions, remotely via the CLI over SSH:

- Configure the access banner (`set banner`)
- Configure the remote session inactivity time before session termination (`set session-timeout`)
- Update the TOE and verify TOE updates prior to installation using digital signature verification (`upload image`)
- Configure local audit behavior (i.e. changes to behaviour when local audit storage space is full; changes to local audit storage size) (`set log`);
- Configure/modify the transmission of audit data to an external IT entity (`set log`);
- Configure the list of TOE-provided services available before an entity is identified and authenticated, per FIA_UIA_EXT.1 (`set respond-to-ping`)
- Manage the cryptographic keys (`upload certificates`, `set ntp authentication symmetric add-key`)
- Configure the list of supported TLS ciphers (`enable cipher-suites`, `disable cipher-suites`)
- Re-enable an administrator account (`enable user`)
- Set the time used for time stamps (`set date`)
- Configure NTP (`set ntp`, `set ntp authentication`)
- Manage the TOE's trust store and designate X.509 v3 certificates as trust anchors (`upload certificate`, `upload ca-certificate`)
- Generate Certificate Signing Request (CSR) and process CA certificate response (`create cert-request`, `upload ca-certificate`)
- Configure the authentication failure parameters for FIA_AFL.1 (`set property server`)
- Manage the trusted public keys database (`add credential cliadmin`)

The TOE also provides the `enable common criteria` CLI command, which performs the following actions to configure the TOE consistent with the evaluated configuration:

- Enforces minimum password length of 15 characters
- Enables lock out for all users after three unsuccessful login attempts. The lockout interval for all users is set to 600 seconds (10 minutes)
- Restricts TLS to TLS v1.2 and disables all other TLS and SSL versions
- Disables disallowed cryptographic algorithms and methods for TLS and SSH communications.

Additionally, the following security management functions can also be performed by the TOE for the **uiadmin** user using the RedSeal Java client/Remote client:

- Configure audit behavior (**`System Settings > Logging`**)
- Manage the TOE's trust store and designate X.509 v3 certificates as trust anchors (**`System Settings > Server Certificate`**)
- Re-enable an administrator account (**`EDIT -> System Settings -> Users`**)

Lastly, the **uiadmin** user is able to manage the warning banner of the Web Beta client (**`Service Admin > Server Settings > Banne`**r). The Web Beta client is otherwise used for non-management functionality.

This aspect of the Security Management security function satisfies FMT_SMF.1.

6.4.3 Management of Security Functions Behavior

The ability to determine and modify the audit behavior is restricted to **cliadmin** and to **uiadmin**.

The ability to perform TOE updates is restricted to **cliadmin** only; this function cannot be performed via any interface besides the CLI.

This aspect of the Security Management security function satisfies FMT_MOF.1/Functions and FMT_MOF.1/ManualUpdate.

6.4.4 Management of TSF Data

The ability to manage TSF data is restricted to the **cliadmin** and **uiadmin**, since the commands for managing TSF data are provided either by the CLI, which only **cliadmin** can access, and the Java client/Remote client and the Web Beta client where such commands are restricted to **uiadmin**. This includes the ability to manage the TOE's trust store by uploading X.509 v3 certificates, CA certificates, and cryptographic keys. **cliadmin** is able to use the **`create cert-request`** CLI command to generate a Certificate Request and upload the certificate returned by the CA in response to the certificate request.

This aspect of the Security Management security function satisfies FMT_MTD.1/CoreData and FMT_MTD.1/CryptoKeys.

6.5 Protection of the TSF

6.5.1 Protection of Administrator Passwords

The TOE protects the passwords for the **cliadmin, uiadmin**, and users with administrative permissions by generating a hash of these passwords using PBKDF2 with SHA-512 and storing the hashed password, rather than storing the password itself or encrypting the password prior to storage. The TOE does not offer any functions that will disclose to any users a plaintext administrative password.

This aspect of the TSF Protection security function satisfies FPT_APW_EXT.1.

6.5.2 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

The TOE does not offer any functions that will disclose to any users a stored cryptographic key. See Section 6.2 for more information about stored keys.

This aspect of the TSF Protection security function satisfies FPT_SKP_EXT.1.

6.5.3 TSF Testing

The TOE performs all self-tests on start-up. These consist of a test to verify the integrity of the TOE firmware/software and cryptographic self-tests that confirm the correct functionality of the cryptographic algorithms implemented by the OpenSSL and BSAFE cryptomodules included in the TOE.

Upon start-up, the TOE will trigger the integrity test to validate the integrity of the main firmware files and RedSeal production code. The test checks the parameters of the Java files against a SHA-512 of the file, the permission of the file, and the ownership of the file. If any of the parameters changed, the system integrity test will fail and the administrative services (Admin server, web server and ssh server) will not startup.

Each cryptomodule performs the following cryptographic self-tests during module initialization:

- Cryptographic known answer tests—for symmetric and one-way cryptographic operations, the module will process known input data and compare it to the pre-computed output for each algorithm (AES KATs, SHA KATs, DRBG KAT, RSA Sign/Verify KAT, ECDSA Sign/Verify KAT) to ensure results are consistent with known answers.
- Pairwise consistency tests—for public key cryptographic operations, the module will perform a cryptographic operation followed by its reverse (e.g., encrypt/decrypt; sign/verify) to ensure that the result of the calculation is the same as the initially-supplied value.

If any of the self-tests fail, the TOE will not progress to a fully operational state. The TOE will be in a maintenance mode where the Admin server, web server and ssh server will not start and either an error message is displayed (for integrity failure) or the result is logged (for cryptographic tests). If the self-tests pass the services will start normally.

When a self-test fails, the appliance is not fully operational and therefore not in its evaluated configuration. In this maintenance mode, an administrator can physically access the appliance where the CLI console will show a message to contact to RedSeal support as a result of an integrity of firmware and software self-test failure. The TOE will log the result of a cryptographic self-test. For example, the log "`CryptoJMode=FIPS140_MODE SelfTestPass=true`" shows the cryptographic self-tests passed.

The self-tests are sufficient to ensure the correct functionality of the TSF as they encompass the cryptographic functionality and the integrity of the entire TOE software/firmware executable code.

This aspect of the TSF Protection security function satisfies FPT_TST_EXT.1.

6.5.4 Trusted Update

The TOE can hold up to three executable images in its file system. One image is designated the "Current" image and one image (which can be the same as the Current image) is designated the "Next" image. The Current image is the image that is currently executing on the TOE, while the Next image is the image that will be loaded for execution at the next reboot of the TOE.

The **cliadmin** can use the `show images` CLI command to display the currently active version of the TOE, and can use the `set next image` CLI command to specify which image will be loaded for execution (and therefore become the Current image) at the next reboot. The **cliadmin** uses the `upload image` CLI command to upload an image to the TOE. An uploaded image is designated the Next image (though this can be changed by the **cliadmin** using the `set next image` CLI command).

Whenever a new GA release is made available, it is published on RedSeal's support portal (https://www.redsealnetworks.com/support—this page requires user authentication). RedSeal publishes appliance images as `.enc` files. The entire package is hashed and signed by RedSeal using SHA256 algorithm and RSA 4096 bit. Customers download images from the support portal and store them on their local filesystem or an HTTPS or SFTP server accessible from the appliance to be upgraded. The **cliadmin** uses the `upload image` CLI command to upload the new image to the appliance. The `upload image` CLI command copies the image and verifies the digital signature of the image after the upload is complete. If the verification is successful, the TOE denotes the uploaded image as the Next image. The Next image become the Current image at the next reboot. That is, a reboot initiates the update process. If verification fails, the TOE rejects the image and does not install it. The **cliadmin** is then able to exit the command and the TOE will automatically delete the unverified image.

This aspect of the TSF Protection security function satisfies FPT_TUD_EXT.1.

6.5.5 Reliable Time Stamps

The TOE comprises the RedSeal Server hardware appliance that includes a hardware-based real-time clock able to provide reliable time stamps for the use of the TOE. The TOE's real-time clock is a Complementary Metal-Oxide Semiconductor that stores the system time and date information. The TOE's embedded OS manages the clock and exposes administrator clock-related functions. The **cliadmin** can use the `show date` CLI command to display the current system date, time and time zone, and can use the `set date` CLI command to set the system date and time to a specified value. The **cliadmin** can use the `set timezone` CLI command to set the time zone of the appliance. The default time zone is Coordinated Universal Time (UTC).

In addition, the TOE can be configured to synchronize its time with an NTP server in the operational environment. The **cliadmin** uses the `set ntp` CLI command to configure the NTP servers to be used by the TOE.

The clock is used for audit record time stamps, measuring session activity for termination, measuring the time an administrator account is locked following consecutive failed authentication attempts, and for cryptographic operations based on time/date, such as checking certificate expiry.

This aspect of the TSF Protection security function satisfies FPT_STM_EXT.1.

6.6 TOE Access

The following methods of administrative access to the TOE are available to the security administrator:

- Remote access to the CLI via SSH v2, restricted to **cliadmin**
- Remote access to the Web Beta client via browser, restricted to **uiadmin** and Web Beta client users with Admin permissions.
- Remote access via the Java client (via standalone Java application or via browser aka "Remote client"), restricted to **uiadmin**

6.6.1 Access Banner

Access banners for each administrative interface are configured independently on the respective interfaces where the banners are shown. The **cliadmin** can use the `set banner` CLI command to configure an advisory notice and consent warning message to be displayed to the user prior to establishment of a CLI session. The `pre-authentication` option ensures the message is displayed after the user enters a username but before the user is prompted to enter a password. On the Java client/Remote client, **uiadmin** can configure the banner for that interface under **`System Settings > Show/Hide`**. On the Web Beta client, **uiadmin** can configure the banner for that interface under **`Service Administration > Server Settings > Banner`**.

This aspect of the TOE Access security function satisfies FTA_TAB.1.

6.6.2 Session Termination

The TOE can be configured to terminate remote interactive sessions after a period of inactivity. The **cliadmin** can use the `set session-timeout` CLI command to configure the session idle timeout value for the remote CLI sessions as a number of minutes. The default value of the CLI session idle timeout is `infinite`, meaning the TSF-initiated termination of inactive remote interactive CLI sessions is disabled by default. This must be configured in the evaluated configuration to specify a maximum value for session

timeout. The **cliadmin** can enable the session idle timeout mechanism for the Web Beta client by using the `set property` command to set the `redseal.srm.https.sessionTimeout` property to `true`. When this property is set to `true`, Web Beta client sessions will be terminated after a period of inactivity as configured by the `set session-timeout` command. Session idle timeout for the Java client/remote client is configured by **uiadmin** under System Settings > Timeout.

Administrators are able to terminate their own interactive sessions by logging out of the TOE. The **cliadmin** logs out of an interactive CLI session by entering the `exit` command. The **uiadmin** logs out of a Java client session by clicking on **File > Exit** or by closing the Java client application. The **uiadmin** logs out of a Legacy/Web Beta client session by clicking on "Log out."

This aspect of the TOE Access security function satisfies FTA_SSL.3 and FTA_SSL.4.

6.7 Trusted Path/Channels

The TOE communicates with the following authorized IT entities:

- Audit server—the TOE can be configured to export its audit records to an external syslog server over TLS. In this case, the TOE acts as a TLS client and initiates the connection to the syslog server. The TOE identifies and authenticates the syslog server by validating the syslog server's X.509 certificate that is presented during the TLS negotiation.

All trusted channel communications are initiated by the TOE.

The TOE supports the following remote access methods:

- Remote access to the CLI via SSH v2, with the TOE acting as the SSH server. This is available to the single CLI user account, **cliadmin**. The **cliadmin** initiates communication to the TSF, using SSH client software on their local workstation. The TSF authenticates the **cliadmin** (either using password or public key) and maintains the trusted path for all remote administration actions until the **cliadmin** terminates the session by exiting the CLI.
- RedSeal Java client/Remote client—the TOE communicates with the RedSeal Java client or Remote client, which provide user-level access to the data modeling and analysis functions of the RedSeal Server, as well as some administrative capabilities. Communication between the Java client and the TOE is via RMI over TLS, with the TOE acting as the TLS server. The TOE identifies and authenticates the Java client by identifying and authenticating the user that attempts to login to the TOE via the Java client. The Remote client is identical in function and presentation to the Java client and uses the same logical interface to access the TOE, except that it is spawned in a web browser process rather than being run as a standalone executable.
- Remote access to the browser-based Web Beta client—**uiadmin** use a browser to communicate with the TOE's Web Beta client. Communication between the Web Beta client and the user's browser is via HTTPS, with the TOE acting as the HTTPS/TLS server. The TOE authenticates the user using password and maintains the trusted path for all remote administration actions until the **uiadmin** terminates the session by exiting the Web Beta client.

All trusted path communications are initiated by the operational environment.

The Trusted Path/Channels security function satisfies FTP_ITC.1 and FTP_TRP.1/Admin.

7 Protection Profile Claims

This ST conforms to the collaborative Protection Profile for Network Devices, Version 3.0e, 06 December 2023 [NDcPP], including the following optional and selection-based SFRs: FAU_STG.1; FAU_STG_EXT.3; FCS_HTTPS_EXT.1; FCS_NTP_EXT.1; FCS_TLSC_EXT.1; FCS_TLSS_EXT.1; FIA_AFL.1; FIA_PMG_EXT.1; FIA_X509_EXT.1/Rev; FIA_X509_EXT.2; FIA_X509_EXT.3; FMT_MOF.1/Functions; FMT_MTD.1/CryptoKeys; and FPT_APW_EXT.1, and to the Functional Package for Secure Shell (SSH), Version 1.0, 13 May 2021 [SSHPKG], including the following selection-based SFRs: FCS_SSHS_EXT.1.

As explained in Section 3, Security Problem Definition, the Security Problem Definition of the [NDcPP] has been included by reference into this ST.

As explained in Section 4, Security Objectives, the ST reproduces the security objectives for the operational environment from [NDcPP].

As explained in Section 5, IT Security Requirements, the SFRs have all been drawn from either [NDcPP] or [SSHPKG] . As such, operations on SFRs already performed in that PP or PKG are not identified in this ST. Rather, the SFRs have been copied from [NDcPP] and [SSHPKG] and any formatting used in that PP or PKG have been removed. Operations performed on SFRs in the writing of this ST are identified in accordance with the conventions described in Section 1.3.

The SARs for the TOE are included by reference from the [NDcPP].

8 Rationale

This Security Target includes by reference the [NDcPP] and [SSHPKG] applicable Security Problem Definition, Security Objectives, and Security Assurance Requirements. The Security Target makes no additions to the [NDcPP] and [SSHPKG] assumptions. Security Functional Requirements have been reproduced verbatim with the Protection Profile and PP-Module operations completed except where refinements were made by the ST author and formatted per the defined convention. Operations on the security requirements follow [NDcPP] and [SSHPKG] application notes and evaluation activities. The Security Target did not add or remove any security requirements. Consequently, [NDcPP] and [SSHPKG] rationales apply and are complete.