RedSeal Server v9.4

Security Target

Version 1.0

2021-06-18

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**
2019-08-05	Leidos	Initial draft.
2019-09-05	Leidos	Second draft following RedSeal review.
2019-10-04	Leidos	Third draft following further RedSeal review and provision of additional information.
2019-10-17	Leidos	Fourth draft following RedSeal responses to open questions.
2019-10-28	Leidos	Updates in response to ASE ETR.
2020-06-17	Leidos	Updates in response to TSS AAR.
2020-07-15	Leidos	Updates for new NIAP Technical Decisions.
2020-08-31	Leidos	Updates for latest TOE version.
2020-12-09	Leidos	Updated to address Evaluation Check-in comments.
2021-05-13	Leidos	Final updates prior to Check-out.
2021-05-18	Leidos	Final clean version ready for Check-out

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 9.4.5 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 ([cPPND] – see Section 1.2 for specific version information). The security functionality specified in [cPPND] 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:

- TOE Description (Section 2)—provides an overview of 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 [cPPND]
- 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 v9.4 Security Target

ST Version: Version 1.0

ST Date: 2021-06-18

TOE Identification: RedSeal Server 9.4.5

TOE Developer: RedSeal, Inc.

Evaluation Sponsor: RedSeal, Inc.

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

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 Protection Profile:

- *collaborative Protection Profile for Network Devices*, Version 2.1, 24 September 2018 [cPPND], including the following optional and selection-based SFRs: FAU_STG.1; FAU_STG.3/LocSpace; FCS_HTTPS_EXT.1; FCS_NTP_EXT.1; FCS_SSHC_EXT.1; FCS_SSHS_EXT.1; FCS_TLSC_EXT.1; FCS_TLSS_EXT.1; FIA_X509_EXT.1/Rev; FIA_X509_EXT.2; FIA_X509_EXT.3; FMT_MOF.1/Functions; and FMT_MOF.1/Services.

The following NIAP Technical Decisions apply to this PP and have been accounted for in the ST development and the conduct of the evaluation:

- TD0396: NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2
- TD0397: NIT Technical Decision for Fixing AES-CTR Mode Tests
- TD0398: NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR
- TD0399: NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)
- TD0400: NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment
- TD0401: NIT Technical Decision for Reliance on external servers to meet SFRs
- TD0402: NIT Technical Decision for RSA-based FCS_CKM.2 Selection
- TD0407: NIT Technical Decision for handling Certification of Cloud Deployments
- TD0408: NIT Technical Decision for local vs. remote administrator accounts
- TD0409: NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication
- TD0410: NIT technical decision for Redundant assurance activities associated with FAU_GEN.1
- TD0411: NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 - Server and client side seem to be confused
- TD0412: NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy
- TD0423: NIT Technical Decision for Clarification about application of RfI#201726rev2
- TD0424: NIT Technical Decision for NDcPP v2.1 Clarification - FCS_SSHC/S_EXT1.5
- TD0425: NIT Technical Decision for Cut-and-paste Error for Guidance AA.
- TD0447: NIT Technical Decision for Using 'diffie-hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7.
- TD0450: NIT Technical Decision for RSA-based ciphers and the Server Key Exchange message.
- TD0451: NIT Technical Decision for ITT Comm UUID Reference Identifier.

- TD0453: NIT Technical Decision for Clarify authentication methods SSH clients can use to authenticate SSH servers.
- TD0475: NIT Technical Decision for Separate traffic consideration for SSH rekey.
- TD0477: NIT Technical Decision for Clarifying FPT_TUD_EXT.1 Trusted Update.
- TD0478: NIT Technical Decision for Application Notes for FIA_X509_EXT.1 iterations.
- TD0480: NIT Technical Decision for Granularity of audit events.
- TD0481: NIT Technical Decision for FCS_(D)TLSC_EXT.X.2 IP addresses in reference identifiers.
- TD0482: NIT Technical Decision for Identification of usage of cryptographic schemes.
- TD0483: NIT Technical Decision for Applicability of FPT_APW_EXT.1.
- TD0484: NIT Technical Decision for Interactive sessions in FTA_SSL_EXT.1 & FTA_SSL.3.
- TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4.
- TD0529: NIT Technical Decision for OCSP and Authority Information Access extension.
- TD0530: NIT Technical Decision for FCS_TLSC_EXT.1.1 5e test clarification.
- TD0531: NIT Technical Decision for Challenge-Response for Authentication.
- TD0532: NIT Technical Decision for Use of seeds with higher entropy.
- TD0533: NIT Technical Decision for FTP_ITC.1 with signed downloads.
- TD0536: NIT Technical Decision for Update Verification Inconsistency.
- TD0538: NIT Technical Decision for Outdated link to allowed-with list.
- TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN.
- TD0570: NiT Technical Decision for Clarification about FIA_AFL.1.
- TD0571: NiT Technical Decision for Guidance on how to handle FIA_AFL.1.
- TD0572: NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers.

The following NIAP Technical Decisions issued for [cPPND] are not applicable to this evaluation, for the reasons stated:

- TD0395: NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2—this TD applies to testing of FCS_TLSS_EXT.2, which is not claimed in this ST.
- TD0535: NIT Technical Decision for Clarification about digital signature algorithms for FPT_TUD.1—this TD provides clarification about digital signature algorithms to support trusted update, but the TOE uses a published hash.

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 [cPPND], 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 [cPPND] 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
SMTP	Simple Message Transfer Protocol
SSH	Secure Shell

Abbreviation	Definition
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
USB	Universal Serial Bus
VGA	Video Graphics Array

2 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 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 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 excluded from the scope of this evaluation.

2.2 TOE Overview

The TOE is RedSeal Server 9.4.5. 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 uses a plugin architecture to import data from monitored networked assets. There are two types of plugins—Communications and Data. Communications plugins provide connectivity to external devices for facilitating data import, supporting various protocols, including SSH, SCP, SFTP, HTTPS, SMB, JDBC, and vendor-specific protocols to access device data directly from the device or from configuration management databases. Data plugins provide parsers for vendor-specific device configuration files and vulnerability scan data. The TOE can monitor most layer 3 network devices with the use of these vendor-specific data plugins. Note the evaluated configuration covers data import using only SSH.

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 locally via a laptop connected directly to a network port, or remotely via SSH. The TOE supports a single CLI user (**cliadmin**) that is equivalent to the Security Administrator role specified in [cPPND].

The RedSeal Client (Java client) also supports an administrator role (**uiadmin**) that can be used to create accounts for non-administrative users of the web interface and additional 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.

Non-administrative users access the TOE via a web interface using a web browser client. This interface does not provide access to any administrative capabilities or security management functions.

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, or provisioned as a virtual appliance image that can be deployed on a virtual platform. The scope of the evaluation covers both the hardware appliance and the virtual appliance deployed on VMware ESXi (other platforms that support the RedSeal Server virtual appliance, such as KVM, Oracle VirtualBox, and Microsoft Hyper-V, are not covered by this 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 (CentOS 7.2-1511) that provides the operating system on which the RedSeal application executes, and Java Runtime Environment (JRE) 8, which supports part of the TOE's cryptographic algorithm implementation. The TOE (through an integrated RSA BSAFE 6.2.1) implements cryptographic algorithms that support secure communication: between the TOE and external IT entities for data collection (SSH); between the TOE and the Java client (TLS); and between the TOE and web clients (HTTPS). The TOE also includes OpenSSL 1.0.2k, which provides the cryptographic algorithms to support SSH connections to the CLI.

2.4 Physical Boundaries

The TOE is available as a hardware appliance (denoted G5b) and as a virtual appliance. The following table summarizes the hardware appliance specifications.

Table 2: G5b Hardware Appliance Specifications

Height	1.7 in (43 mm)
Width	17.2 in (437 mm)
Depth	23.5 in (597 mm)
Weight	44 lbs (20 kg)
Temperature	32 – 122 degrees F (0 – 50 degrees C)
Humidity (noncondensing)	5 – 95 %
Voltage	100-240V, 8.5A-3.8A, 50-60 Hz
Processor	Intel Xeon E5-2637 v4 (Broadwell microarchitecture)
RAM	128 GB
Disk storage	1.75 TB usable storage (6 x 1 TB configured as RAID 10)
Power	Dual hot plug redundant (1 + 1) 700W

The virtual appliance is supported on VMware ESXi v6.5. The server on which ESXi and the TOE virtual appliance are installed require a minimum 64 GB RAM and 4 processor cores. RedSeal recommends 128 GB RAM and 8 processor cores for performance equivalent to the G5b hardware appliance. Evaluation testing of the virtual appliance was conducted on an AMD Ryzen Threadripper 1950X (Zen) processor.

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

- Syslog server for external storage of exported audit records
- Laptop directly connected to a network port for local administrator access to CLI
- Administrative workstation equipped with SSH client software for remote administrator access to CLI.

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

- NTP server
- SMTP server, for receiving email notifications from the TOE
- SFTP server, to support upload, backup, and restore operations
- Workstation with web browser for accessing the Web UI
- Workstation running Windows or Mac OS X with Java 8 and web browser for downloading, installing and running the Java client.

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 is able to generate audit records of security relevant events, including the events specified in [cPPND]. 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 80% 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 local and remote access (via SSH) to a Command Line Interface (CLI) and remote access (protected by TLS) using the Java client to support interactive administrator sessions. There is also a browser-based Web UI available for non-administrative users to interact with the TOE.

The TOE provides a local password-based authentication mechanism for all users and enforces a minimum length for passwords. The TOE will deny remote access to a user after a configurable number of consecutive failed 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 published hash.

The TOE provides a CLI to access its security management functions. Administrators can access the CLI locally via a laptop connected directly to a network port and remotely using SSH. Additionally, some security management functions are accessible via 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 published hash.

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

2.5.6 TOE Access

The TOE will terminate local and 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) and remote non-administrative users using HTTPS (for access to the Web UI).

The TOE is able to protect transmission of audit records to an external audit server using TLS. It uses SSH to connect to external IT entities for the purpose of data collection, to support building its model of the network. It also protects communication with the Java client 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 9.4
- RedSeal User Guide, Version 9.4
- RedSeal 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 [cPPND]. 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 [cPPND] has presented a Security Problem Definition appropriate for network infrastructure devices, and as such is applicable to the RedSeal Server.

4 Security Objectives

The [cPPND] 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.

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 [cPPND]. As such, operations on SFRs already performed in that PP are not identified here. Rather, the SFRs have been copied from [cPPND] and any formatting used in that PP has 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 [cPPND].

5.1 Extended Requirements

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

- FAU_STG_EXT.1—Protected Audit Event Storage
- FCS_HTTPS_EXT.1—HTTPS Protocol
- FCS_NTP_EXT.1—NTP Protocol
- FCS_SSHC_EXT.1—SSH Client 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_UAU_EXT.2—Password-Based Authentication Mechanism
- 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
- FTA_SSL_EXT.1—TSF-Initiated Session Locking

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

Requirement Class	Requirement Component
	FAU_STG_EXT.1 Protected audit event storage
	FAU_STG.3/LocSpace 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_SSHC_EXT.1 SSH client 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: Identification and authentication	FIA_AFL.1 Authentication failure management
	FIA_PMG_EXT.1 Password management
	FIA_UIA_EXT.1 User identification and authentication
	FIA_UAU_EXT.2 Password-based authentication mechanism
	FIA_UAU.7 Protected authentication feedback
	FIA_X509_EXT.1/Rev X.509 certificate validation
	FIA_X509_EXT.2 X.509 certificate authentication
	FIA_X509_EXT.3 X.509 certificate requests
FMT: Security management	FMT_MOF.1/Functions Management of security functions behavior
	FMT_MOF.1/ManualUpdate Management of security functions behavior
	FMT_MOF.1/Services Management of security functions behavior
	FMT_MTD.1/CoreData 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 pre-shared, symmetric and private keys)
	FPT_STM_EXT.1 Reliable time stamps
	FPT_TST_EXT.1 TSF testing

Requirement Class	Requirement Component
	FPT_TUD_EXT.1 Trusted update
FTA: TOE access	FTA_SSL_EXT.1 TSF-initiated session locking
	FTA_SSL.3 TSF-initiated termination
	FTA_SSL.4 User-initiated termination
	FTA_TAB.1 Default TOE access 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 user account shall be logged if individual user accounts are required for Administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- [***no other actions***];

d) Specifically defined auditable events listed in Table 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	None.	None.
FAU_STG.3/LocSpace	Low storage space for audit events.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_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_SSHC_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1	Failure to establish a TLS session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure
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 the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None.
FMT_MOF.1/Services	None.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_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_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.

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.
[***TOE shall consist of a single standalone component that stores audit data locally***].

FAU_STG_EXT.1.3 The TSF shall [***overwrite previous audit records according to the following rule: [the oldest 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.

5.2.1.5 Action in Case of Possible Audit Data Loss (FAU_STG.3/LocSpace)

FAU_STG.3.1/LocSpace The TSF shall generate a warning to inform the Administrator if 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)

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

- ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;***
- ***ECC schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;***
- ***FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1***
- ***FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3***

].

5.2.2.2 Cryptographic Key Establishment (FCS_CKM.2)

Note, this SFR has been modified in accordance with TD0402.

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

- ***RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1";***
- ***Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";***
- ***Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";***
- ***Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3;***

].

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

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

- ***RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits],***

]
that meet the following: [

- ***For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3***

].

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.

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

5.2.2.9 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.

5.2.2.10 SSH Client Protocol (FCS_SSHC_EXT.1)

Note, this SFR has been modified in accordance with TD0398, TD0424, TD0453, and TD0475.

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) [***4251, 4252, 4253, 4254, 4344, 5656, 6668***].

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

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [**35,000**] bytes in an SSH transport connection are dropped.

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

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

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [***hmac-sha1, hmac-sha2-256, hmac-sha2-512***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7 The TSF shall ensure that [***diffie-hellman-group14-sha1, ecdh-sha2-nistp256***] and [***ecdh-sha2-nistp384, ecdh-sha2-nistp521***] are the only allowed key exchange methods used for the SSH protocol.

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

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

5.2.2.11 SSH Server Protocol (FCS_SSHS_EXT.1)

Note, this SFR has been modified in accordance with TD0398 and TD0424.

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) [***4251, 4252, 4253, 4254, 4344, 5656, 6668***].

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

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [**35,000**] bytes in an SSH transport connection are dropped.

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

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

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [***hmac-sha1*, *hmac-sha2-256*, *hmac-sha2-512***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [***diffie-hellman-group14-sha1*, *ecdh-sha2-nistp256***] and [***diffie-hellman-group14-sha256*, *diffie-hellman-group16-sha512*, *ecdh-sha2-nistp384*, *ecdh-sha2-nistp521***] are the only allowed key exchange methods used for the SSH protocol.

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

5.2.2.12 TLS Client Protocol (FCS_TLSC_EXT.1)

Note, this SFR has been modified in accordance with TD0481.

FCS_TLSC_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:
[
- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246***
- ***TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***

].

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifiers of the following types: [***identifiers defined in RFC 6125***] are matched to reference identifiers.

FCS_TLSC_EXT.1.3 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [***Not implement any administrator override mechanism***].

FCS_TLSC_EXT.1.4 The TSF shall [***present the Supported Elliptic Curves Extension with the following NIST curves: [secp256r1, secp384r1, secp521r1]***] in the Client Hello.

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), TLS 1.1 (RFC 4346)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:
[

- ***TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246***
- ***TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***

].

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

FCS_TLSS_EXT.1.3 The TSF shall [***perform RSA key establishment with key size [2048 bits]; generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves***].

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 (any), CTR_DRBG (AES)***].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[1] software-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)

Note, this SFR has been modified in accordance with TD0408.

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 remote session using any authentication method that involves a password until [action to unlock the Administrator account by entering the account unlock command at the CLI] is taken by an Administrator; prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed***].

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.

5.2.3.4 Password-Based Authentication Mechanism (FIA_UAU_EXT.2)

Note, this SFR has been modified in accordance with TD0408.

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

5.2.3.5 Protected Authentication Feedback (FIA_UAU.7)

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

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

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

5.2.3.7 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 [***[users authenticating with CAC]***].

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***].

5.2.3.8 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 Security Functions Behavior (FMT_MOF.1/Services)

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

5.2.4.4 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.

5.2.4.5 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 locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***hash comparison***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [
  - ***Ability to start and stop services***;

- ***Ability to configure audit behavior*;**
- ***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 configure the cryptographic functionality*;**
- ***Ability to re-enable an Administrator account*;**
- ***Ability to set the time which is used for time-stamps*;**
- ***Ability to configure NTP*;**
- ***Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors*;**
- ***Ability to import X.509v3 certificates to the TOE's trust stor*e;**

].

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

are satisfied.

5.2.5 Protection of the TSF (FPT)

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

Note, this SFR has been modified in accordance with TD0483.

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***].

5.2.5.4 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 [***published hash***] prior to installing those updates.

5.2.5.5 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [**cryptographic known answer tests, pairwise consistency tests**].

5.2.6 TOE Access (FTA)

5.2.6.1 TSF-Initiated Session Locking (FTA_SSL_EXT.1)

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

5.2.6.2 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.

5.2.6.3 User-Initiated Termination (FTA_SSL.4)

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

5.2.6.4 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 [***SSH, TLS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[data collection from external devices; remote administration via the RedSeal client; email notifications to SMTP server over TLS; upload, backup and restore operations to external SFTP server]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [**export of audit records to external syslog server, data collection from external devices, sending email notifications to SMTP server, and upload, backup and restore operations with external SFTP server**].

5.2.7.2 Trusted Path (FTP_TRP.1/Admin)

FTP_TRP.1.1/Admin The TSF shall be capable of using [***SSH, 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 security assurance requirements for the TOE are included by reference from the [cPPND].

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
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 use of the identification and authentication mechanism, including the origin
- 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, or by identifying (for SSH public keys) the hostname or IP address of the server associated with the public key.
- Resetting passwords, including identification of the relevant user account
- Any failure to establish an HTTPS, TLS, or SSH session, including the reason for the failure
- Configuration and removal of NTP server, including the identity of the server
- The number of consecutive failed authentication attempts has been reached, including the origin of the failed attempts
- Unsuccessful attempts to validate an X.509 certificate, including the reason for failure
- Addition, replacement and removal of a certificate in the trust store, including identification of the certificate
- Initiation of a manual update of the TOE and the result (success, failure) of the update
- Discontinuous changes to the system time, including both the old and new values for the time, and the origin of the attempt to change the time (both success and failure)
- Termination of local and remote interactive sessions by the session locking mechanism
- Termination by a user of their own interactive session
- Initiation and termination of trusted channels and failure of trusted channel functions, including initiator and target of failed initiation attempts

- Initiation, termination and failure of trusted path functions.

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; and the outcome of the event (e.g., success or failure). 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 or virtual appliance that is able to store generated audit records locally (i.e., on the appliance or on the server hosting the virtual 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
- Analyzer—contains records of data collection events, including date, time, name of the event, name of the credential used, and the communication method used. The analyzer log also contains records of the results of analysis events
- 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 frequency or size (but not both—size is used by default). The **cliadmin** can configure logs to be rotated daily, weekly, or monthly, or when they reach a maximum size. The size must be in the range of 1,000KB 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 80% 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).

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

6.2 Cryptographic Support

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

- OpenSSL—the TOE includes an OpenSSL implementation (`openssl-1.0.2k-12.el7.x86_64`) used in conjunction with OpenSSH (`openssh-7.4p1-16.el7.x86_64`) to support SSH connections to the CLI (SSH server functionality).
- RSA BSAFE—the TOE includes RSA BSAFE 6.2.1, which it uses to support secure communications with external IT entities for data collection (SSH client), backup and restore operations to an external SFTP server (SSH client), email notifications to an SMTP server (TLS client), the Java client (TLS server), and web user access (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.

Table 7: Cryptographic Functions Implemented by OpenSSL

Functions	Standards	Certificates
Asymmetric Key Generation (FCS_CKM.1)		
RSA (2048 bits)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	A1058 RSA KeyGen (FIPS 186-4)
ECDSA (P-256, P-384, P-521 curves)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	A1058 ECDSA KeyGen (FIPS186-4)
DSA (2048 bits)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1	A1058 DSA KeyGen (FIPS186-4)
Key establishment (FCS_CKM.2)		
Elliptic curve-based scheme	NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	A1058 KAS-ECC Component
Finite field-based scheme	NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	A1058 KAS-FFC Component
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)	A1058 AES-CBC A1058 AES-CTR
Digital signature generation and verification (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (2048 bit modulus)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS2v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	A1058 RSA SigGen (FIPS 186-4) A1058 RSA SigVer (FIPS 186-4)

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	A1058 SHA-1 A1058 SHA2-256 A1058 SHA2-384 A1058 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"	A1058 HMAC-SHA-1 A1058 HMAC-SHA2-256 A1058 HMAC-SHA2-512
Deterministic random bit generation (FCS_RBG_EXT.1)		
Counter DRBG	ISO/IEC 18031:2011	A1058 Counter DRBG

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

Table 8: Cryptographic Functions Implemented by RedSeal Server

Functions	Standards	Certificates
Asymmetric Key Generation (FCS_CKM.1)		
RSA (2048 bits)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	C1889 RSA KeyGen (FIPS186-4)
ECDSA (P-256, P-384, P-521 curves)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	C1889 ECDSA KeyGen (FIPS186-4), ECDSA KeyVer (FIPS186-4)
DSA (2048 bits)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1	C1889 DSA KeyGen (FIPS186-4)
Key establishment (FCS_CKM.2)		
RSA-based scheme	RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"	n/a
Elliptic curve-based scheme	NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	C1889 KAS-ECC Component
Finite field-based scheme	NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	C1889 KAS-FFC Component

Functions	Standards	Certificates
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)	C1889 AES-CBC C1889 AES-GCM C1889 AES-CTR
Digital signature generation and verification (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (2048 bit modulus)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS2v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	C1889 RSA SigGen (FIPS186-4), RSA SigVer (FIPS186-4)
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	C1889 SHA-1 C1889 SHA2-256 C1889 SHA2-384 C1889 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"	C1889 HMAC-SHA-1 C1889 HMAC-SHA2-256 C1889 HMAC-SHA2-384 C1889 HMAC-SHA2-512
Deterministic random bit generation (FCS_RBG_EXT.1)		
Hash DRBG	ISO/IEC 18031:2011	C1889 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 186-4, "Digital Signature Standard (DSS)".

The TOE performs cryptographic hashing services using SHA-1, SHA-256, SHA-284, 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)
- to verify the integrity of uploaded update images
- for NTP authentication.

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 Hash DRBG provided by its BSAFE module to generate random bits for use with other BSAFE algorithms. The TOE seeds the Hash DRBG with 256 bits of entropy obtained from the `/dev/random` pseudo-device.

The TOE instantiates the Counter DRBG (AES) provided by OpenSSL to generate random bits for use with other OpenSSL algorithms. The TOE does not instantiate the OpenSSL Counter DRBG until after the BSAFE cryptomodule (including its DRBG) is fully initialized. The TOE seeds the Counter DRBG with 256 bits of entropy obtained from the `/dev/urandom` pseudo-device.

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, in accordance with Appendix B.3 of FIPS PUB 186-4, "Digital Signature Standard (DSS)". The RSA keys are used in support of SSH public key authentication and TLS server authentication.

The TOE generates ECC asymmetric key pairs over NIST curves P-256, P-384, and P-521, in accordance with Appendix B.4 of FIPS PUB 186-4, "Digital Signature Standard (DSS)". The ECDSA keys are used in support of SSH and TLS key exchange.

The TOE generates FFC asymmetric key pairs with cryptographic key sizes (modulus) of 2048 and 4096 bits, in accordance with Appendix B.1 of FIPS PUB 186-4, "Digital Signature Standard (DSS)". The DSA keys are used in support of SSH key exchange. The 2048 bit keys support Diffie-Hellman group 14 with SHA-256, while the 4096 bit keys support Diffie-Hellman group 16 with SHA-512.

The TOE generates FFC asymmetric key pairs using Diffie-Hellman group 14, in accordance with RFC 3526, Section 3. These keys are used in support of SSH key exchange.

The TOE acts as both a sender and recipient in the SSH key establishment schemes. It acts as a client for communication with external monitored devices and backup and restore operations to an external SFTP server, and 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 sending email notifications to an external SMTP server, and as a server for remote administration via the RedSeal client and the web interface.

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
RSA	FCS_TLSC_EXT.1	Audit server; SMTP server
RSA	FCS_TLSS_EXT.1	Administration
ECDH	FCS_TLSC_EXT.1	Audit server; SMTP server
ECDH	FCS_TLSS_EXT.1	Administration
ECDH	FCS_SSHC_EXT.1	Data collection from external devices Backup and restore to SFTP server
ECDH	FCS_SSHS_EXT.1	Administration
DH (Group 16)	FCS_SSHS_EXT.1	Administration
DH (Group 14)	FCS_SSHC_EXT.1	Data collection from external devices Backup and restore to SFTP server
DH (Group 14)	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 (2048 bit modulus)	Generated by TOE. Used to authenticate the TOE in TLS and SSH sessions. Stored on disk in PKCS #12 keystore.
EC DH private key (P-256, P-384, P-512 NIST curves)	Generated by TOE. Used in TLS and SSH key exchange. Stored in RAM.
DH private key	Generated by TOE. Used in 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 and sending email notifications to an external SMTP server
- TLS as a server—the TOE acts as a TLS server supporting inbound communications from the Java client
- HTTPS—in conjunction with TLS, the TOE supports the use of HTTPS for remote user access via the Web UI
- SSH-2 as a client—the TOE acts as an SSH client when performing data collection from external devices and performing upload, backup, and restore operations to an external SFTP server
- 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 v1.1 and TLS v1.2 and the following TLS cipher suites when acting as a TLS client:

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_RSA_WITH_AES_128_CBC_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 NIST curves: secp256r1; secp384r1; and secp521r1. This is done by default and is not configurable.

The TOE establishes the reference identifier for the external audit server based on the hostname (fully qualified domain name – FQDN) of the audit server configured by the **cliadmin** using the `set log` CLI command. The TOE establishes the reference identifier for the external SMTP server based on the hostname (FQDN) of the SMTP server configured by the **uiadmin** using the **Email** tab of the **System Settings** dialog on the RedSeal Java client. In both cases, 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 supports wildcards for peer authentication. Certificate pinning is not supported.

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 Java client. In its evaluated configuration (after the **cliadmin** has executed the `enable common criteria` CLI command), the TOE supports TLS v1.1 and v1.2 and the following TLS cipher suites when acting as a TLS server:

- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_RSA_WITH_AES_128_CBC_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 v1.1 and will reject ClientHello messages that specify support only for TLS v1.0 or versions of SSL.

When the TOE negotiates a cipher suite that uses ECDHE_RSA as its key exchange algorithm, it sends a Server Key Exchange message that specifies the supported NIST curve, the ECDH public key, and the associated domain parameters. In conformance with RFC 5246, the TOE does not send the Server Key Exchange message when negotiating a cipher suites that uses RSA for the key exchange algorithm.

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

6.2.5.3 HTTPS

The TOE uses HTTPS to protect communications between itself and remote users accessing the Web UI. 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 client when performing data collection from external devices and upload, backup, and restore operations to an external SFTP server, and 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, and 6668. It supports both public key and password-based authentication as described in RFC 4252.

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

- Encryption algorithms—aes128-cbc, aes256-cbc, aes128-ctr, and aes256-ctr
- Public key authentication algorithm—ssh-rsa
- Data integrity MAC algorithms—hmac-sha1, hmac-sha2-256, and hmac-sha2-512
- Key exchange methods—diffie-hellman-group14-sha1, ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521.

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
- Public key authentication algorithm—ssh-rsa
- Data integrity MAC algorithms—hmac-sha1, hmac-sha2-256, and hmac-sha2-512
- Key exchange methods—diffie-hellman-group14-sha1, ecdh-sha2-nistp256, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, ecdh-sha2-nistp384, and ecdh-sha2-nistp521.

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. It has the following effects:

- disables the following encryption algorithms: aes192-cbc; and aes192-ctr
- disables the following MAC algorithms: hmac-md5; and umac-64@openssh.com
- disables the following key exchange methods for both SSH client and SSH server: curve25519-sha256; curve25519-sha256@libssh.org; diffie-hellman-group-exchange-sha256; diffie-hellman-group18-sha512; diffie-hellman-group-exchange-sha1; and diffie-hellman-group1-sha1
- additionally disables the following key exchange methods for SSH client: diffie-hellman-group14-sha256; diffie-hellman-group16-sha512.

The TOE ensures that packets greater than 35,000 bytes 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. The TOE initiates a rekey when any of the thresholds is reached, whichever is hit first.

The TOE's SSH client implementation authenticates the identity of SSH servers using a local database that associates each host name with its corresponding public host key.

This aspect of the Cryptographic Support security function satisfies FCS_SSHC_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, locally via a laptop connected directly to a network port, or 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 using the Java client, remotely via TLS. The **uiadmin** is able to create additional accounts that can access the TOE using the Java client or the Web UI.

In order to log in, the user must provide an identity and also authentication data that matches that identity configured on the TOE. The TOE implements a password-based mechanism to authenticate the **cliadmin** when accessing the TOE locally. Users logging on to **cliadmin** remotely via SSH can be configured to authenticate with a public key, so that it is not necessary for multiple users to share the **cliadmin** password. Users accessing the Web UI can be defined by the Subject DN associated with their smart card certificate for certificate-based authentication.

The TOE supports the following logon methods:

- Local connection to the CLI, using username and password—the user enters the **cliadmin** user name and associated password when prompted. The TOE does not display the password as it is entered, but instead displays '*' characters. If the logon is successful, the user is presented with the CLI command prompt. The user terminates the local connection to the CLI by entering the `exit` command at the CLI prompt.
- 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** user name 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** user name. 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.

- 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)
  - User name—RedSeal user account ID
  - Password—password for the entered RedSeal account
- Remote connection to Web UI via HTTPS, using username and password—the user uses browser software on their local workstation to initiate an HTTPS connection to port 443 on the TOE. The user is presented with a login dialog where the username and associated password are entered. If the logon is successful, the user is presented with the Home tab in the GUI.
- Remote connection to Web UI via HTTPS, using username and certificate—the user uses browser software on their local workstation to initiate an HTTPS connection to port 10443 on the TOE. The user is presented with a dialog to select the certificate to be used for authentication. If the Subject DN in the certificate is associated with the user's account on the TOE and the TOE is able to validate the user's certificate, the logon is successful and the user is presented with the Home tab in the GUI.

This aspect of the Identification and Authentication security function satisfies FIA_UIA_EXT.1, FIA_UAU_EXT.2, and FIA_UAU.7.

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 from the Java client, the Web UI, and remotely to the CLI (local access to the CLI is not blocked). The default lockout period is 600 seconds (10 minutes).

The **cliadmin** is able to configure individually 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, 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 using the Java client or via the Web UI is performed using the Java client or Web UI.

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 or user authentication, where the certificate is stored on the user's CAC. 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[1] 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 TLS servers (i.e., external syslog server, external SMTP server) and of users authenticating with CAC. 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.

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.

[1] 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.

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 on the CLI, locally via a laptop connected directly to a network port, or remotely via SSH
- **uiadmin**—performs administrative tasks using the Java client, remotely via TLS.

Security management commands provided by the CLI and the Java 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, access to security management functions is restricted to **uiadmin** and to user accounts created with Admin permissions.

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 via the CLI, both locally and remotely:

- Configure the access banner (`set banner`)
- Configure the session inactivity time before session termination or locking (`set session-timeout`)
- Update the TOE and verify TOE updates prior to installation using hash comparison (`upload image`)
- Configure the authentication failure parameters (`set property server`)
- Start and stop services (`startup`, `shutdown`, `enable autostart`, `disable autostart`)
- Configure audit behavior (`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`)
- Configure cryptographic functionality (`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 ca-certificate`)
- Import X.509 v3 certificates to the TOE's trust store (`upload certificate`).

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 v1.1 or 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 (or Java client user with Admin permissions) using the Java client:

- Update the TOE and verify TOE updates prior to installation using hash comparison (**Admin > Server**, select **Upload Image**)
- 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**)
- Import X.509 v3 certificates to the TOE's trust store (**System Settings > Server Certificate**).

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

6.4.3 Management of Security Functions Behavior

The ability to start and stop TOE services and determine and modify the behavior of handling of audit data is restricted to the **cliadmin**. The ability to perform TOE updates is restricted to **cliadmin** and to **uiadmin** (or Java client users with Admin permissions).

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

6.4.4 Management of TSF Data

The ability to manage TSF data is restricted to the **cliadmin** and **uiadmin** (or Java client user with Admin permissions), since the commands for managing TSF data are provided either by the CLI, which only **cliadmin** can access, or the Java client, where such commands are restricted to **uiadmin** and Java client users with Admin permissions. This includes the ability to manage the TOE's trust store by uploading X.509 v3 certificates and CA certificates.

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

6.5 Protection of the TSF

6.5.1 Protection of Administrator Passwords

The TOE protects the passwords for the **cliadmin** and Java client users (including **uiadmin**) 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 cryptographic self-tests that confirm the correct functionality of the cryptographic algorithms implemented by the OpenSSL and BSAFE cryptomodules included in the TOE. 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 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.

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 and images for virtual appliances as `.ova` files. Along with each release, RedSeal publishes a SHA-512 hash value. 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 computes a SHA-512 hash value over the image after the upload is complete. The command prompts the **cliadmin** to enter the SHA-512 hash value published with the TOE image, which it compares to its calculated hash to verify the integrity of the image. If the verification is successful, the TOE denotes the uploaded image as the Next image. If verification fails, the TOE prompts the **cliadmin** to re-enter the hash value. If the image is corrupted, so the **cliadmin** is unable to verify the image, the **cliadmin** is able to exit the command and the TOE will automatically delete the unverified image.

Alternatively, **uiadmin** or a Java client user with Admin permissions can use the **Upload Image** command from the **Admin > Server** page of the Java client to upload and verify a TOE update 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 and the RedSeal Server virtual appliance image installed on a physical server, both of which include 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:

- Local access to the CLI using a laptop directly connected to a network port, restricted to **cliadmin**
- Remote access to the CLI via SSH v2, restricted to **cliadmin**
- Remote access via the Java client, restricted to **uiadmin** and Java client users with Admin permissions. Note, this does not constitute a remote interactive session, since the communication is between the Java client (an external IT entity) and the TOE.

The TOE also provides remote access for non-administrative users to the Web UI via HTTPS.

6.6.1 Access Banner

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 an administrative user session, for each

method of administrative access. 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.

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 both local and 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. The **cliadmin** can enable the session idle timeout mechanism for the Web UI by using the `set property` command to set the `redseal.srm.https.sessionTimeout` property to `true`. When this property is set to `true`, Web UI sessions will be terminated after a period of inactivity as configured by the `set session-timeout` command.

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.

This aspect of the TOE Access security function satisfies FTA_SSL_EXT.1, 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.
- Monitored external devices—the TOE connects to external devices on the network in order to collect data from them for the purpose of building its network model. In this case, the TOE acts as an SSH client and initiates the connection to the monitored external device. The TOE identifies and authenticates the monitored external device by validating its SSH server host key.
- Email server—the TOE can be configured to communicate with a network mail server in order to send email notifications to configured recipients. In the evaluated configuration, this connection is configured to communicate SMTP traffic over TLS.
- File server—the TOE communicates with external file servers for the purposes of backing up and restoring RedSeal server data and uploading image updates. In the evaluated configuration, communications with external file servers use SFTP, which runs over SSH. In this circumstance, the TOE acts as the SSH client.
- RedSeal Java client—the TOE communicates with the RedSeal Java client, which provides 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 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.

- Remote access to Web UI via HTTPS, with the TOE acting as the HTTPS server. The user initiates communication to the TSF, using a web browser on their local workstation. The TSF authenticates the user (either using password or certificate associated with a CAC) and maintains the trusted path for all remote user actions until the user terminates the session by exiting the Web UI. Note, the Web UI does not provide any security management functions within the scope of FMT_SMF.1. However, some administrative actions can be performed via the Java client, which is considered an external IT entity from the perspective of the TOE.

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 2.1, 24 September 2018 [cPPND] and including the following optional and selection-based SFRs: FAU_STG.1; FAU_STG.3/LocSpace; FCS_HTTPS_EXT.1; FCS_NTP_EXT.1; FCS_SSHC_EXT.1; FCS_SSHS_EXT.1; FCS_TLSC_EXT.1; FCS_TLSS_EXT.1; FIA_X509_EXT.1/Rev; FIA_X509_EXT.2; FIA_X509_EXT.3; FMT_MOF.1/Functions; and FMT_MOF.1/Services.

As explained in Section 3, Security Problem Definition, the Security Problem Definition of the [cPPND] 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 [cPPND].

As explained in Section 5, IT Security Requirements, the SFRs have all been drawn from [cPPND]. As such, operations on SFRs already performed in that PP are not identified in this ST. Rather, the SFRs have been copied from [cPPND] and any formatting used in that PP has 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 [cPPND].

8 Rationale

This ST includes by reference the [cPPND] Security Problem Definition and SARs and reproduces the security objectives for the Operational Environment. The ST makes no additions to the [cPPND] assumptions. [cPPND] SFRs have been reproduced with the Protection Profile operations completed. Operations on the SFRs follow [cPPND] application notes and assurance activities. Consequently, [cPPND] rationale applies but is incomplete. The TOE Summary Specification rationale below serves to complete the rationale required for the security target.

8.1 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification (TSS), describes a security function of the TOE. Each description identifies the SFRs that are covered by that description and, as such, provides the rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The security functions work together to satisfy all of the security functional requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This section, in conjunction with the TSS, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TSS are all necessary for the required security functionality in the TSF. Table 12: Security Functions vs. Requirements Mapping summarizes the relationship between security requirements and security functions.

Table 12: Security Functions vs. Requirements Mapping

Specification	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FAU_GEN.1	X						
FAU_GEN.2	X						
FAU_STG.1	X						
FAU_STG_EXT.1	X						
FAU_STG.3/LocSpace	X						
FCS_CKM.1		X					
FCS_CKM.2		X					
FCS_CKM.4		X					
FCS_COP.1/DataEncryption		X					
FCS_COP.1/SigGen		X					
FCS_COP.1/Hash		X					
FCS_COP.1/KeyedHash		X					

Specification	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FCS_HTTPS_EXT.1		X					
FCS_NTP_EXT.1		X					
FCS_RBG_EXT.1		X					
FCS_SSHC_EXT.1		X					
FCS_SSHS_EXT.1		X					
FCS_TLSC_EXT.1		X					
FCS_TLSS_EXT.1		X					
FIA_AFL_EXT.1			X				
FIA_PMG_EXT.1			X				
FIA_UIA_EXT.1			X				
FIA_UAU_EXT.2			X				
FIA_UAU.7			X				
FIA_X509_EXT.1/Rev			X				
FIA_X509_EXT.2			X				
FIA_X509_EXT.3			X				
FMT_MOF.1/Functions				X			
FMT_MOF.1/ManualUpdate				X			
FMT_MOF.1/Services				X			
FMT_MTD.1/CoreData				X			
FMT_SMF.1				X			
FMT_SMR.2				X			
FPT_APW_EXT.1					X		
FPT_SKP_EXT.1					X		
FPT_TST_EXT.1					X		
FPT_TUD_EXT.1					X		
FPT_STM_EXT.1					X		
FTA_SSL_EXT.1						X	
FTA_SSL.3						X	
FTA_SSL.4						X	
FTA_TAB.1						X	
FTP_ITC.1							X

Specification	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FTP_TRP.1/Admin							X