Red Hat
Certificate System 8.1
Security Target

Revision 1.0
February 15, 2012

Prepared for:
Red Hat, Inc.

1801 Varsity Drive
Raleigh, North Carolina 27606

Prepared By:

Science Applications International Corporation

6841 Benjamin Franklin Driver
Columbia, MD 21046

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

LIST OF TABLES

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

LIST OF FIGURES

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is Red Hat Certificate System 8.1 (RHCS 8.1) provided by Red Hat, Inc. RHCS 8.1 is a certificate issuing and management system offering the following general services to users and/or administrators:

- Certificate Enrollment,
- Certificate Renewal,
- Certificate Revocation,
- Certificate Retrieval,
- Certification and Certificate Revocation List (CRL) Management,
- Key Archival and Retrieval Service,
- Token Management System, and
- Online Certificate Status Protocol (OCSP) Responder Service.

1.1 Security Target, TOE and CC Identification

ST Title – Red Hat Certificate System 8.1 Security Target

ST Version – Revision 1.0

ST Date – February 15, 2012

TOE Identification – Red Hat Certificate System 8.1

CC Identification – Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, September 2007 (CCv3.1).

1.2 Conformance Claims

This TOE conforms to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 3.1, Revision 2, September 2007.
  - Part 2 extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 3.1, Revision 2, September 2007.
  - Part 3 conformant
  - Evaluation Assurance Level 4 (EAL 4) augmented with ALC_FLR.2
- Certificate Issuing and Management Components (CIMC) In Basic Robustness Environments Protection Profile (PP), Version 1.0, April 27, 2009 (CIMC-BR-PP).

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the Security Target.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

1.3.1 Conventions

The following conventions have been applied in this document:

- All requirements in this ST are reproduced relative to the requirements defined in CC v3.1r2.
- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - For operations performed while incorporating requirements from the CIMC-BR-PP the following conventions were used:
        - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter in parenthesis placed at the end of the component. For example FDP_ACC.1(a) and FDP_ACC.1(b) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
        - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that in cases where a selection operation is combined with an assignment operation and the assignment is null, the assignment operation is simply deleted leaving on the completed selection to identify the combination of operations.
        - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
        - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.2 Terminology and Acronyms

See section 11 (Glossary of terms) and section 12 (Acronyms).

1.4 Security Target Overview and Organization

The Red Hat Certificate System 8.1 (RHCS 8.1) Target of Evaluation (TOE) is a Certificate Management System offering a wide range of certificate related services. This Security Target describes the RHCS 8.1 TOE, intended environments, security objectives, security requirements (for the TOE and IT environment), security functions, Protection Profile claims, and all necessary rationale. This information is organized the following additional sections:

- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)
- Access control policies (Section 9)
- Strength of Function (SoF) Requirements (Section 10)

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

- Glossary of terms (Section 11)
- Acronyms (Section 12)

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

2. TOE Description

The target of evaluation (TOE) is Red Hat Certificate System version 8.1 (RHCS 8.1).

2.1 TOE Overview

RHCS 8.1 provides a security framework to guarantee the identity of users and ensure privacy of communications. RHCS 8.1 issues and manages X.509v3 certificates needed to handle strong authentication, single sign-on and secure communications. RHCS 8.1 handles all the major functions around the certificate lifecycle simplifying enterprise-wide deployment and adoption. Customizable registration allows RHCS 8.1 to adapt to virtually any enterprise security policy.

2.1.1 Intended Environment

Red Hat Certificate System 8.1 (RHCS 8.1) is appropriate for environments where risks and consequences of data disclosure and loss of data integrity are moderate. Certificate Issuing and Management Component (CIMC) In Basic Robustness Environments Protection Profile (CIMC-BR-PP) requires integrity controls to ensure data is not modified. A CIMC-BR-PP -conformant CIMC, such as RHCS 8.1, includes protections to protect against someone with physical access to the components and includes assurance requirements to ensure the CIMC is functioning securely.

The CIMC-BR-PP requires protection against malicious authorized users by requiring at least three distinct roles. At a minimum, one role will be responsible for account administration, key generation, and audit configuration; a second role will be responsible for issuing and revoking certificates; and a third role responsible for maintaining the audit logs. The CIMC-BR-PP requires two-party control of private key export and additional auditing of import and export of secret and private keys and requests for information. Cryptographic modules responsible for long-term private key protection or for signing certificates or certificate status information must be validated to FIPS 140-2 Level 3. Finally, there is increased public key protection and digital signatures are required on all messages.

While the CIMC-BR-PP requires only Evaluation Assurance Level 2 (EAL2) augmented with flaw remediation requirements (ALC_FLR.2), EAL 4 (augmented with ALC_FLR.2) has been adopted as the overall assurance level for RHCS 8.1. An EAL 4 evaluation includes an analysis supported by “gray box” testing, selective independent confirmation of the developer test results, and evidence of a developer search for obvious vulnerabilities. EAL 4 also includes an analysis of the design of the modules of the TOE, and a subset of the implementation. Testing is supported by an independent search for vulnerabilities. ALC_FLR.2 addresses the need for the remediation of flaws even after a product has been evaluated. Both higher assurance (e.g., EAL 4) and flaw remediation are important for many customers to have a moderate degree of assurance in the products they purchase.

In its intended environment, RHCS 8.1 protects its own functions through the implementation of security management and access control policies as detailed later in this ST. The interfaces offered by RHCS 8.1 have been carefully designed to require user identification and authentication, where appropriate, so that access to its functions and data can be controlled and audited. RHCS 8.1 depends on its host operating system for an execution environment, network communication stack, as well as protection of the RHCS 8.1 data and code stored within the operating system (e.g., files) both while executing and at rest.

2.2 TOE Architecture

The RHCS 8.1 TOE is an operating system application written in Java, C++, C, and Perl using associated network (Network Security Services; NSS) and java (Java Security Services; JSS) security service libraries. The RHCS 8.1 TOE is designed to integrate with a directory server such as Red Hat Directory Server to provide an internal data store and a HTTP engine (Tomcat or Apache, depending on the TOE component) to provide a network interface. The underlying JSS and NSS are designed to support the use of hardware devices that perform standards-oriented cryptographic operations. All of the components represent a RHCS 8.1 system. A RHCS 8.1 system is designed to

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

be hosted within a RHEL 5.6+, with Security-Enhanced Linux (SELinux) policies specifically designed to protect the subsystems of the TOE, and to be connected to networks, including the Internet, and to offer these services using standard HTTP/SSL protocols.

A RHCS 8.1 system is composed of the following key components (the first of which is the TOE and the others are key supporting components in the TOE's environment):

- Certificate System (CS)

  The CS includes five configurable subsystems that work together to manage enterprise PKI deployments, including:

  - Certificate Authority (CA) - the subsystem that provides certificate management functionality for issuing, renewing, revoking, and publishing certificates and creating and publishing Certificate Revocation Lists (CRLs).
  - Data Recovery Manager (DRM) - an optional subsystem that provides private encryption key storage and retrieval. Also, in a Token Management System setup, generates key pairs for the clients when server-side key generation option is turned on.
  - Online Certificate Status Protocol (OCSP) Manager - an optional subsystem that provides OCSP responder services, based on stored CA's CRLs to distribute the load for certificate status verification.
  - Token Key Service (TKS) - manages one or more master keys required to set up secure channels from the tokens directly to the token processing system. The secure channels provided by TKS allows Global Platform compliant smart cards (tokens) to be identified with high level of confidence and subsequently communicate securely with the RHCS servers for operations such as certificate enrollments, renewals, server-side key generation requests, key archival and recovery, etc.
  - Token Processing System (TPS) - one unique function of the TPS is to provide communication between Global Platform-compliant smart cards and the RHCS systems by means of *APDU* (Application Protocol Data Unit). It provides the registration authority functionality in the token management infrastructure and with the assistance of the TKS, establishes secure channels between the smart cards and the back-end subsystems.

  The CS subsystems (CA, DRM, OCSP Manager, TKS, and TPS) are highly integrated with each other depending on the deployment scenario. OCSP and CA instances work together on CRL publishing and certificate verification. CA and DRM instances work together for key recovery and archival. Smart card tokens, processed through the Enterprise Security Client (ESC) user interface, are managed by the TPS. The TPS, however, is designed to work with at least two essential subsystem instances, a TKS to manage shared secrets between the tokens and TMS and a CA to process certificate enrollment operations. A TPS can also be configured to use a DRM for server-side key generation and key archival and recovery, with the assistance of TKS to deliver private keys securely to the tokens (smart cards).

  The CA, DRM, OCSP Manager, and TKS are implemented in Java, utilize a Tomcat HTTP engine (see below), and share a common framework (also written in Java) for management, logging, authentication, access control, self tests, and notifications. The TPS is written as a native RHEL 5.6+ C++ application and utilizes an Apache HTTP engine.

- HTTP Engines (Tomcat (*for CA, DRM, OCSP Manager, and TKS*) & Apache (*for TPS*))

  The web engine provides the HTML-based UI (presentation) and HTTP-based protocol handling. It does not perform authentication and authorization other than providing and/or enforcing SSL. It performs basic certificate validation and delegates all the application-specific authentication and authorization to CS via a callback mechanism.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

- Internal Database (Red Hat Directory Server - RHDS 8.1)

  The internal database stores information such as certificates, requests, officer/administrator information, and other information such as access control information. The CS communicates with the internal database securely through SSL client authentication.

The following architectural diagrams show the interactions between various CS configurations and various internal and external systems. Internally, the CS communicates with an internal database where certificate records, request records, system user records are stored. The CS also accesses the cryptographic operations (directly or indirectly) via NSS. Externally, the HTTP engine manages the presentation-level interaction between the CS and users including end-users, security officers, and administrators. The CS may optionally publish certificates to a corporate directory server.

In addition to the HTTP Engine and Internal Database, the CS also relies on access to processing capabilities, file storage, as well as hardware cryptographic modules provided by its IT environment.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

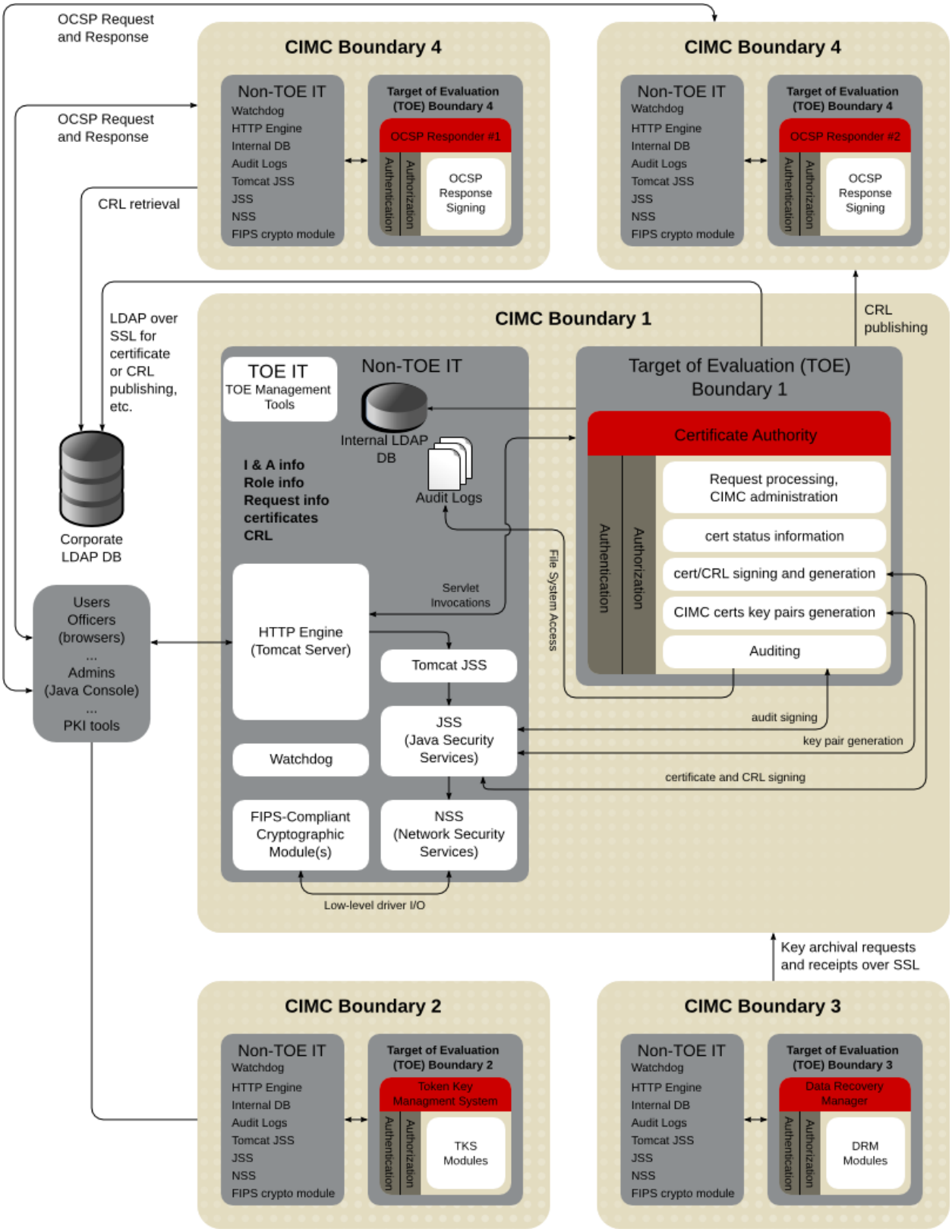

Figure 1 RHCS 8.1 System Overview

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

The Non-TOE IT environments are similar among all CIMC boundaries. Please refer to CIMC Boundary 1 in Figure 1 and Figure 2 to see complete details for all other Non-TOE IT within other CIMC boundaries. Figure 2 shows the TPS component and its connections to the other RHCS 8.1 components.

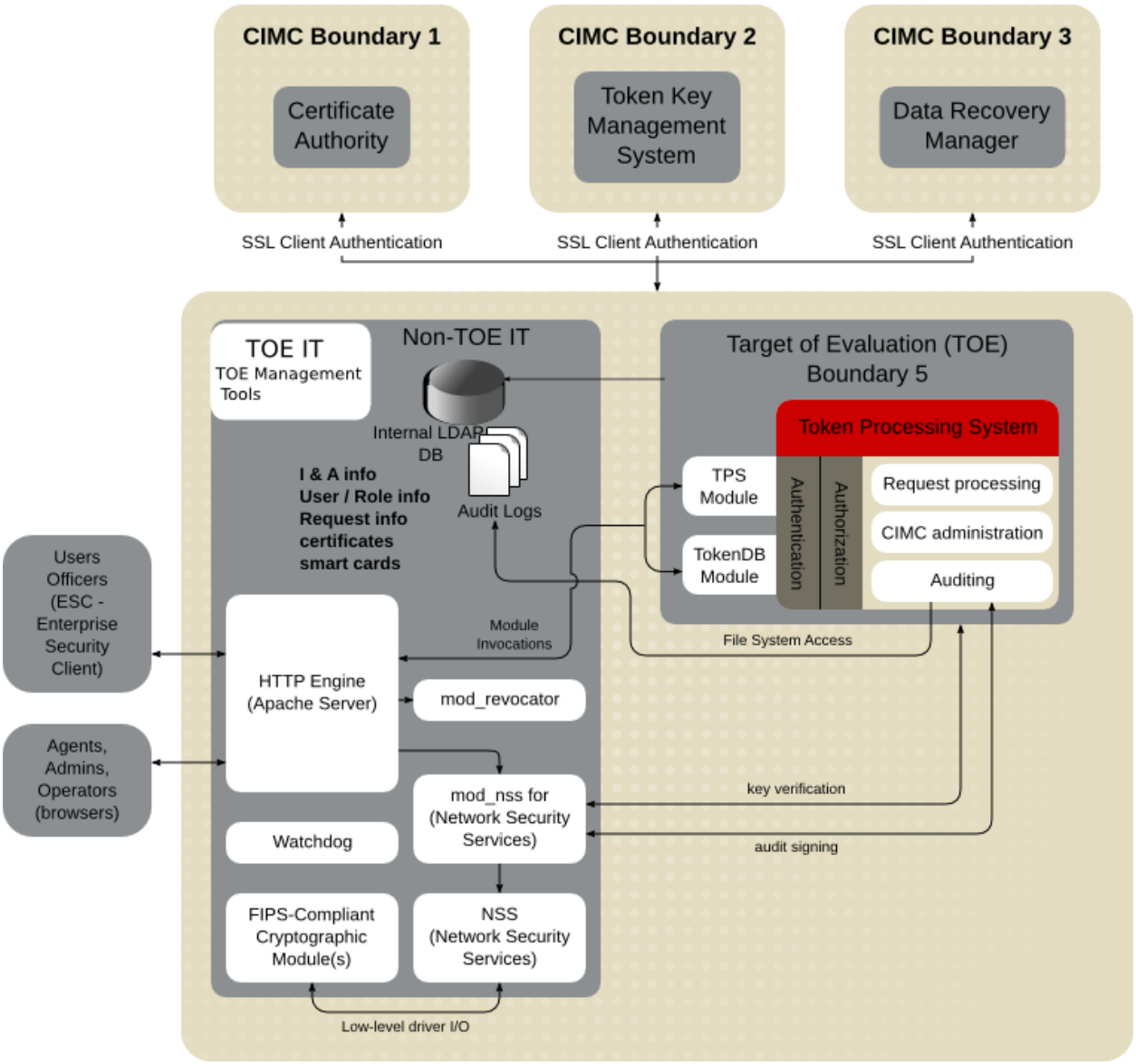

Figure 2 Token Management System

While a complete RHCS 8.1 *system* includes all of the components within the CIMC boundaries indicated in Figure 1 and Figure 2, the RHCS 8.1 *TOE* includes the components within the labeled TOE Boundaries. Specifically, the TOE consists of the CA, OCSP Manager, DRM, TKS, and TPS subsystems (some of which are optional depending on the PKI application). The RHCS 8.1 TOE includes a Java GUI-based administration tool called the 'Console' that is used for administrative tasks such as managing users and maintaining the (CA, OCSP Manager, DRM, and TKS) subsystems and performing daily operational and managerial duties for those subsystems. Additionally, the RHCS

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

8.1 TOE includes a number of command-line utilities (see the Red Hat Certificate System 8.1 Command-Line Tools Guide for a complete list and more information), for example:

- The AuditVerify tool is used to verify that signed audit logs were signed with the private signing key and that the audit logs have not been compromised. Auditors can verify the authenticity of signed audit logs using the AuditVerify tool. This tool uses the public key of the signed audit log signing certificate to verify the digital signatures embedded in a signed audit log file. The tool response indicates either that the signed audit log was successfully verified or that the signed audit log was not successfully verified. An unsuccessful verification warns the auditor that the signature failed to verify, indicating the log file may have been tampered with (compromised).
- The PIN Generator generates unique PINs for end-entity entries in an LDAP directory. The tool stores these PINs as hashed values in the same directory against the corresponding user entries. It also copies the PINs to a text file so that the PINs can be sent to the end entities.
- The TKS utility manages keys, including keys stored on tokens, the TKS master key, and related keys and databases. It offers the following functions: deleting a key from a token; inputting shares to generate a new transport key; displaying the key check value (KCV) of the specified key; listing a specified key or all keys; generating a new master key; creating a new key database; changing the key database password; renaming a symmetric key; listing all security modules; generating a new transport key; unwrapping a wrapped master key; and wrapping a new master key.
- The CMC Request, Enrollment, Responses, and Revocation utilities to create CMC requests request from PKCS #10 or CRMF requests; to sign a certificate request with an agent's certificate; parse CMC responses received by the utility; and sign a revocation request with an agent's certificate, respectively.
- The CRMF Pop Request utility is a tool to send a Certificate Request Message Format (CRMF) request to a Certificate System CA with the request encoded with proof of possession (POP) data that can be verified by the CA server. If a client provides POP information with a request, the server can verify that the requester possesses the private key for the new certificate.
- The HTTP Client utility sends a CMC request (created with the CMC Request utility) or a PKCS #10 request to a CA.
- The OCSP request utility creates an OCSP request conforming to RFC 2560, submits it to the OCSP server, and saves the OCSP response in a file.
- The PKCS #10 utility generates a public key pair in the security database, constructs a PKCS#10 certificate request with the public key, and outputs the request to a file.
- The Revocation Automation utility sends revocation requests to the CA agent interface to revoke certificates.

2.3 Physical Boundaries

As depicted in Figure 1, the TOE exists as a collection application programs interacting with other components to implement its security functions. The TOE applications run within an IT environment based on RHEL 5.6+ (with configured SELinux policies) and including a Java runtime environment (with JSS/NSS libraries), a HTTP Engine (i.e., Tomcat or Apache), and a directory server (e.g., Red Hat Directory Server) and watchdog daemon.

The TOE supports LDAP interfaces and also HTTP-based interfaces. The LDAP interfaces are used to connect to the internal LDAP Server (e.g., Red Hat Directory Server) used by RHCS 8.1 exclusively as a private data store, and also to connect to a Corporate LDAP server for publishing purposes, if configured. The HTTP-based interfaces allow users, administrators, agents, auditors, and operators to connect to RHCS 8.1 to access its security functions and to manage RHCS 8.1.

Since the TOE is a collection of application programs, its logical and physical boundaries largely coincide. The TOE requires basic execution, data storage support, and network connectivity services from its IT environment. The external interfaces are limited to LDAP (over SSL), HTTP/SSL, and the use of command-line utility programs.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

LDAP connections are supported only when initiated by RHCS 8.1. The HTTP/SSL interfaces are used to offer functions via service-oriented web pages to RHCS 8.1 users, officers, agents, auditors, and administrators. The command-line utility programs make use of these other interfaces; data files (e.g. for configuration or audit review); and in some cases do not interact with the rest of the TOE at all.

Note that administrative functions (for the CA, DRM, OCSP, and TKS subsystems) are performed using a console application included with RHCS 8.1. This application interacts with the CS using HTTP/SSL, but instead of using XML/HTML it also uses proprietary name/value pairs to better facilitate the administrator functions available. The TPS subsystem is managed via changes in configuration files (using RHEL OS tools) and through a web browser using HTTP/SSL.

The components of the TOE include:

- Primary Certificate System components:
    - Certificate Authority (CA)
    - Data Recovery Manager (DRM)
    - Online Certificate Status Protocol (OCSP) Manager
    - Token Key Service (TKS)
    - Token Processing System (TPS)
- Command-line tools:
    - PKI setup tools
        - pkiarch/pkidist/pkiflavor/pkiname/pkiperl
        - pkicreate/pkiremove/pkicommon
        - pkisilent
        - p7tool
        - pkihost
    - TOE management tools
        - AtoB (ASCII to Binary)
        - AuditVerify
        - BtoA (Binary to ASCII)
        - CMCEnroll
        - CMCRequest
        - CMCResponse
        - CMCRevoke
        - CRMFPopClient (CRMF Pop Request)
        - ExtJoiner (Extension Joiner)
        - GenExtKeyUsage (Key Usage Extension)
        - GenIssuerAltNameExt (Issuer Alternative Name Extension)
        - GenSubjectAltNameExt (Subject Alternative Name Extension)
        - HttpClient

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

  - OCSPClient
  - PKCS10Client (PKCS #10 Client)
  - PKCS12Export
  - PrettyPrintCert (Pretty Print Certificate)
  - PrettyPrintCrl (Pretty Print Certificate Revocation List)
  - TokenInfo
  - setpin (PIN Generator)
  - sslget
  - tkstool
  - revoker
- Guidance documents:
  - Red Hat Certificate System 8.1 Release Notes
  - Red Hat Certificate System 8.1 Managing Smart Cards with the Enterprise Security Client
  - Red Hat Certificate System 8.1 Administrator's Guide
  - Red Hat Certificate System 8.1 Agent's Guide
  - Red Hat Certificate System 8.1 Command-Line Tools Guide
  - Red Hat Certificate System 8.1 Deploy and Install Guide
  - Red Hat Certificate System 8.1 Using End User Services

The components of the TOE environment include:

- Red Hat Enterprise Linux (RHEL) 5.6+ – provides basic execution, data storage support, and network connectivity services.
- Open Java Development Kit (JDK)/Java Runtime Environment (JRE) 1.6+.
- Java Security Services (JSS) 4.6+ – provide security services to applications (e.g., encryption).
- Network Security Services (NSS) 3.12+ – provide security services to applications (e.g., encryption).
- Tomcat 5.5.23+ (and) and Apache 2.2.3+ – provide web-based (HTTP/HTTPS) interfaces being clients and the TOE.
- Tomcatjss, mod_nss (1.0.8+), and mod_revocator (1.0.3+) (shipped with RHEL) – provide network security services to applications (e.g., encryption).
- Red Hat Directory Server 8.2+ – provides the internal directory (database storage) for the TOE.
- Firefox 3.x+ – provides a browser for web services access.
- Hardware Security Module (HSM) – Thales nCipher netHSM – provides the FIPS-certified cryptographic services related to certificate management for the TOE.
- Enterprise Security Client (ESC) – provides the client to access token services available via the TPS.
- mozldap-tools (6.0.5+) and perl-Mozilla-LDAP (1.5.2-4+) – provides useful ldap tools (search, modify, delete).
- nss-tools (3.12+) – provides tools used to debug and develop NSS applications.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

- Nuxwdog (1.0.0-14+) – provides watchdog daemon services that can stop and start the server.
- PKI Console - java-based GUI tool used for administration of CA, DRM, OCSP, and TKS instances.

2.3.1 TOE Protection

The RHCS 8.1 TOE is designed to protect itself and also to rely on supporting protections from other components. At a high level, the TOE utilizes a separate and distinct hardware cryptographic engine for critical cryptographic operations; the TOE is designed to make effective use of SELinux security mechanisms to protect itself and its underlying data and executables; the TOE command-line tools do not operate on or modify live TOE data, but rather use the documented security interfaces of the TOE to interact with the TOE; the TOE security functions are modular to isolate them from potential errors in other components; and the TOE interfaces are well-defined and restricted using a common certificate-based access control mechanism to distinguish among and limit the functions of administrator roles.

The TOE protects itself primarily using its identification & authentication and access control functions. With these functions, it ensures that users are properly authenticated and they are authorized to perform the functions made available by the TOE. Users that cannot be authenticated or that are not authorized will be denied access to applicable TOE functions.

The TOE relies on the components identified above for security and non-security functions. The primary security functions involve protecting the TOE as it is executing or at rest within its host, in facilitating secure inter-component communication, and to provide FIPS-compliant cryptographic services.

The host operating system and Java implementation are relied upon to provide a distinct and separate execution environment for the TOE applications. In order to make effective use of the operating system all RHCS 8.1components are packaged utilizing standard Red Hat RPM package management. As such, whenever the TOE components are installed, they are stored with “root” user and group ownership and utilize standard Linux directory, file, and executable UNIX permissions. When an RHCS 8.1 TOE instance is generated from these installed components, a “pkiuser’ user and group identifier is used for ownership of *most* portions of the installed instances. The notable exceptions are (1) that an instance's start/stop script is ONLY granted “root” ownership with read/write/execute permission available only to root and (2) that the signed audit log files contained under the signedAudit directory contain a group privilege of "pkiaudit" to allow separation of roles between auditors and administrators. Files owned by “pkiuser” containing potentially sensitive information (e. g., log files, configuration files such as CS.cfg, and NSS security database files) contain no privileges for "other" users (e.g., file permissions of 00660 or 00600). Also, the entire contents of each PKI instance’s signed audit directory are not accessible to "other" users. In practice, access to the “root” account is limited to administrators and the “pkiuser” account is configured so that it is not used by any human user, but rather is used by TOE components.

While previous versions of the TOE were designed to operate in an unconfined SELinux domain, a SELinux policy was created specifically to enhance the protection of RHCS 8.1. This policy includes the following characteristics:

1. The files and directories delivered by for each of the subsystems are labeled with specific SELinux contexts (*pki_ca_exec_t*, *pki_ca_var_lib_t*, *pki_ca_var_log_t*, etc. for a CA for example).
2. The ports used by the each subsystem are labeled with specific SELinux contexts (*pki_ca_port*, *pki_tps_port,* etc.)
3. The CS subsystem processes are also constrained to run within specific SELinux domains (*pki_ca_t*, *pki_ra_t*, *pki_ocsp_t*, etc.). When processes are started, they start in the *unconfined_t* domain, but transition into their assigned domain.
4. Each SELinux domain has rules written to specify the actions that are authorized for the domain. As an example, the *pki_ca_t* domain has rules written to allow write-access files with context *pki_ca_var_log_t*. Moreover, it has rules to allow processes running within the domain to connect to ports of type *pki_ca_port* (as well as others).
5. All accesses not specified in the policy are denied.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Ultimately, the operating system with SELinux extensions is configured to protect the TOE and its stored data using the core access control mechanisms and SELinux domain protection mechanisms.

The TOE also relies on its security providers (JSS/NSS) and web engines primarily to facilitate secure (SSL/HTTPS) communications between TOE components and also with TOE clients. While the TOE can support a number cipher suites with RSA key exchange, the following are recommended since they are FIPS compliant: AES and SHA-2 Message Authentication (with 128- or 256-bit keys) and Triple DES and SHA-2 Message Authentication (with 168-bit keys).

Finally, the TOE depends on a HSM that has been FIPS certified to provide the underlying cryptographic support necessary to allow the TOE to securely fulfill the certificate management expectations of the CIMC-BR-PP. The HSM is accessed via a corresponding library installed on the host operating system. The HSM stores critical keys so that they are not externally accessible. It provides access to its embedded keys in order to generate new keys, encrypt/decrypt data, produce signatures, etc. In practice, the TOE is the sole user or client of the HSM attached directly to its host operating system.

2.4 Logical Boundaries

The RHCS 8.1 TOE is designed to offer security functions generally expected of Certificate Issuing and Management Systems. While administrators of the TOE may have access to available command-line utilities, other users are limited to services offered via the web-based HTTP/HTTPS interfaces.

The RHCS 8.1 TOE offers the security functions summarized in the following subsections, each of which is described in more detail in section 6, “TOE Summary Specification”.

2.4.1 Identification & Authentication

RHCS 8.1 ensures that users are identified and authenticated before they can access any other security relevant services.

2.4.2 Access Control

RHCS 8.1 provides the ability to define an access control list for each service it provides. These access control lists are used to ensure that users can only access services they have been authorized to use.

2.4.3 Security Management

RHCS 8.1 uses the access control functions to control the actions of administrative personnel. In order to accomplish this, predefined access control lists are assigned to the applicable services.

2.4.4 Security Audit

RHCS 8.1 has the capability to audit security relevant events. Audit records are generated when audit events occur, including the responsible user, date, time, and other details. Audit records are collected into audit buffers that are signed, to protect against possible tampering of the audit records, and then copied into non-volatile audit logs.

2.4.5 Remote Data Entry & Export

RHCS 8.1 protects data import and export operations using SSL sessions and secure channels in the case of TMS.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

2.4.6 Key Management

RHCS 8.1 includes a number of key management functions. In particular, RHCS 8.1 protects security critical keys and other information by either encrypting it or storing it within a hardware cryptographic module. RHCS 8.1 also uses digital signatures when appropriate to ensure the integrity of key management related information.

2.4.7 Certificate Management

RHCS 8.1 includes a number of certificate management functions. In particular, RHCS 8.1 allows administrators to control, limit, or mandate values in certificates, certificate revocation lists (CRLs), and online certificate status protocol (OCSP) responses that are generated.

2.4.8 Strength of Functions

RHCS 8.1 is designed to make appropriate use of a FIPS 140-2 certified Hardware Security Module (HSM) for critical cryptographic operations.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

3. Security Problem Definition

This section includes the following:

- Secure usage assumptions,
- Threats, and
- Organizational security policies.

This information provides the basis for the security objectives specified in Section 4, "Security Objectives" and the security functional requirements for the TOE specified in Section 5, "IT Security Requirements".

3.1 Secure Usage Assumptions

The usage assumptions are organized in three categories: personnel (assumptions about administrators and users of the system as well as any threat agents), physical (assumptions about the physical location of the TOE or any attached peripheral devices), and connectivity (assumptions about other IT systems that are necessary for the secure operation of the TOE).

3.1.1 Personnel Assumptions

A.Auditors Review Audit Logs

Audit logs are required for security-relevant events and must be reviewed by the Auditors.

A.Authentication Data Management

An authentication data management policy is enforced to ensure that users change their authentication data at appropriate intervals and to appropriate values (e.g., proper lengths, histories, variations, etc.) (Note: this assumption is not applicable to biometric authentication data.)

A.Competent Administrators, Operators, Officers and Auditors

Competent Administrators, Operators, Officers and Auditors will be assigned to manage the TOE and the security of the information it contains.

A.CPS

All Administrators, Operators, Officers, and Auditors are familiar with the certificate policy (CP) and certification practices statement (CPS) under which the TOE is operated.

A.Disposal of Authentication Data

Proper disposal of authentication data and associated privileges is performed after access has been removed (e.g., job termination, change in responsibility).

A.Malicious Code Not Signed

Malicious code destined for the TOE is not signed by a trusted entity.

A.Notify Authorities of Security Issues

Administrators, Operators, Officers, Auditors, and other users notify proper authorities of any security issues that impact their systems to minimize the potential for the loss or compromise of data.

A.Social Engineering Training

General users, administrators, operators, officers and auditors are trained in techniques to thwart social engineering attacks.

A.Cooperative Users

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Users need to accomplish some task or group of tasks that require a secure IT environment. The users require access to at least some of the information managed by the TOE and are expected to act in a cooperative manner.

3.1.2 Physical Assumptions

A.Communications Protection

The system is adequately physically protected against loss of communications i.e., availability of communications.

A.Physical Protection

The TOE hardware, software, and firmware critical to security policy enforcement will be protected from unauthorized physical modification.

3.1.3 Connectivity Assumptions

A.Operating System

The operating system has been selected to provide the functions required by this CIMC to counter the perceived threats for the appropriate Security Level identified in this family of PPs.[1]

3.2 Threats

The threats are organized in four categories: authorized users, system, cryptography, and external attacks.

3.2.1 Authorized Users

T.Administrative errors of omission

Administrators, Operators, Officers or Auditors fail to perform some function essential to security.

T.User abuses authorization to collect and/or send data

User abuses granted authorizations to improperly collect and/or send sensitive or security-critical data.

T.User error makes data inaccessible

User accidentally deletes user data rendering user data inaccessible.

T.Administrators, Operators, Officers and Auditors commit errors or hostile actions

An Administrator, Operator, Officer or Auditor commits errors that change the intended security policy of the system or application or maliciously modify the system's configuration to allow security violations to occur.

3.2.2 System

T.Critical system component fails

Failure of one or more system components results in the loss of system critical functionality.

T.Malicious code exploitation

An authorized user, IT system, or hacker downloads and executes malicious code, which causes abnormal processes that violate the integrity, availability, or confidentiality of the system assets.

T.Message content modification

[1] This assumption has been copied directly from the CIMC PP. In the context of this ST, "appropriate Security Level identified in this family of PPs" reflects Security Level 3 as represented by this ST.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

A hacker modifies information that is intercepted from a communications link between two unsuspecting entities before passing it on to the intended recipient.

T.Flawed code

A system or applications developer delivers code that does not perform according to specifications or contains security flaws.

3.2.3 Cryptography

T.Disclosure of private and secret keys

A private or secret key is improperly disclosed.

T.Modification of private/secret keys

A secret/private key is modified.

T.Sender denies sending information

The sender of a message denies sending the message to avoid accountability for sending the message and for subsequent action or inaction.

3.2.4 External Attacks

T.Hacker gains access

A hacker masquerades as an authorized user to perform operations that will be attributed to the authorized user or a system process or gains undetected access to a system due to missing, weak and/or incorrectly implemented access control causing potential violations of integrity, confidentiality, or availability.

T.Hacker physical access

A hacker physically interacts with the system to exploit vulnerabilities in the physical environment, resulting in arbitrary security compromises.

T.Social engineering

A hacker uses social engineering techniques to gain information about system entry, system use, system design, or system operation.

3.3 Organization Security Policies

P.Authorized use of information

Information shall be used only for its authorized purpose(s).

P.Cryptography

FIPS-approved or NIST-recommended cryptographic functions shall be used to perform all cryptographic operations.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

4. Security Objectives

This section includes the security objectives including security objectives for the TOE, security objectives for the environment, and security objectives for both the TOE and environment.

4.1 Security Objectives for the TOE

This section includes the security objectives for the TOE, divided among four categories: authorized users, system, cryptography, and external attacks.

4.1.1 Authorized Users

O.Certificates

The TSF must ensure that certificates, certificate revocation lists, and certificate status information are valid.

4.1.2 System

O.Preservation/trusted recovery of secure state

Preserve the secure state of the system in the event of a secure component failure and/or recover to a secure state.

4.1.3 Cryptography

O.Non-repudiation

Prevent user from avoiding accountability for sending a message by providing evidence that the user sent the message.

4.1.4 External Attacks

O.Control unknown source communication traffic

Control (e.g., reroute or discard) communication traffic from an unknown source to prevent potential damage.

4.2 Security Objectives for the Environment

This section specifies the security objectives for the environment.

O.Administrators, Operators, Officers and Auditors guidance documentation

Deter Administrator, Operator, Officer or Auditor errors by providing adequate documentation on securely configuring and operating the CIMC.

O.Auditors Review Audit Logs

Identify and monitor security-relevant events by requiring auditors to review audit logs on a frequency sufficient to address level of risk.

O.Authentication Data Management

Ensure that users change their authentication data at appropriate intervals and to appropriate values (e.g., proper lengths, histories, variations, etc.) through enforced authentication data management (Note: this objective is not applicable to biometric authentication data.)

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

O.Communications Protection

Protect the system against a physical attack on the communications capability by providing adequate physical security.

O.Competent Administrators, Operators, Officers and Auditors

Provide capable management of the TOE by assigning competent Administrators, Operators, Officers and Auditors to manage the TOE and the security of the information it contains.

O.CPS

All Administrators, Operators, Officers and Auditors shall be familiar with the certificate policy (CP) and the certification practices statement (CPS) under which the TOE is operated.

O.Disposal of Authentication Data

Provide proper disposal of authentication data and associated privileges after access has been removed (e.g., job termination, change in responsibility).

O.Installation

Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security.

O.Malicious Code Not Signed

Protect the TOE from malicious code by ensuring all code is signed by a trusted entity prior to loading it into the system.

O.Notify Authorities of Security Issues

Notify proper authorities of any security issues that impact their systems to minimize the potential for the loss or compromise of data.

O.Physical Protection

Those responsible for the TOE must ensure that the security-relevant components of the TOE are protected from physical attack that might compromise IT security.

O.Social Engineering Training

Provide training for general users, Administrators, Operators, Officers and Auditors in techniques to thwart social engineering attacks.

O.Cooperative Users

Ensure that users are cooperative so that they can accomplish some task or group of tasks that require a secure IT environment and information managed by the TOE.

O.Lifecycle security

Provide tools and techniques used during the development phase to ensure security is designed into the CIMC. Detect and resolve flaws during the operational phase.

O.Repair identified security flaws

The vendor repairs security flaws that have been identified by a user.

O.Cryptographic functions

The TOE must implement approved cryptographic algorithms for encryption/decryption, authentication, and signature generation/verification; approved key generation techniques and use validated cryptographic modules. (Validated is defined as FIPS 140-2 validated.)

O.Operating System

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

The operating system used is validated to provide adequate security, including domain separation and non-bypassability, in accordance with security requirements recommended by the National Institute of Standards and Technology.

O.Periodically check integrity

Provide periodic integrity checks on both system and software.

O.Security roles

Maintain security-relevant roles and the association of users with those roles.

O.Social Engineering Training

Provide training for general users, Administrators, Operators, Officers and Auditors in techniques to thwart social engineering attacks.

O.Sufficient backup storage and effective restoration

Provide sufficient backup storage and effective restoration to ensure that the system can be recreated.

O.Validation of security function

Ensure that security-relevant software, hardware, and firmware are correctly functioning through features and procedures.

O.Trusted Path

Provide a trusted path between the user and the system. Provide a trusted path to security-relevant (TSF) data in which both end points have assured identities.

4.3 Security Objectives for both the TOE and the Environment

This section specifies the security objectives that are jointly addressed by the TOE and the environment. While normally security objectives are assigned to the TOE or the environment, but not both, the CIMC-BR-PP introduces this set of jointly addressed security objectives. Sections 5 and 6 of the CIMC-BR-PP identify the corresponding requirements for the TOE and its intended environment. That information serves to differentiate the specific expectations for the TOE and its environment relative to these objectives.

O.Configuration Management

Implement a configuration management plan. Implement configuration management to assure identification of system connectivity (software, hardware, and firmware), and components (software, hardware, and firmware), auditing of configuration data, and controlling changes to configuration items.

O.Data import/export

Protect data assets when they are being transmitted to and from the TOE, either through intervening untrusted components or directly to/from human users.

O.Detect modifications of firmware, software, and backup data

Provide integrity protection to detect modifications to firmware, software, and backup data.

O.Individual accountability and audit records

Provide individual accountability for audited events. Record in audit records: date and time of action and the entity responsible for the action.

O.Integrity protection of user data and software

Provide appropriate integrity protection for user data and software.

O.Limitation of administrative access

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Design administrative functions so that Administrators, Operators, Officers and Auditors do not automatically have access to user objects, except for necessary exceptions. Control access to the system by Operators and Administrators who troubleshoot the system and perform system updates.

O.Maintain user attributes

Maintain a set of security attributes (which may include role membership. access privileges, etc.) associated with individual users. This is in addition to user identity.

O.Manage behavior of security functions

Provide management functions to configure, operate, and maintain the security mechanisms.

O.Object and data recovery free from malicious code

Recover to a viable state after malicious code is introduced and damage occurs. That state must be free from the original malicious code.

O.Procedures for preventing malicious code

Incorporate malicious code prevention procedures and mechanisms.

O.Protect stored audit records

Protect audit records against unauthorized access, modification, or deletion to ensure accountability of user actions.

O.Protect user and TSF data during internal transfer

Ensure the integrity of user and TSF data transferred internally within the system.

O.Require inspection for downloads

Require inspection of downloads/transfers.

O.Respond to possible loss of stored audit records

Respond to possible loss of audit records when audit trail storage is full or nearly full by restricting auditable events.

O.Restrict actions before authentication

Restrict the actions a user may perform before the TOE authenticates the identity of the user.

O.Security-relevant configuration management

Manage and update system security policy data and enforcement functions, and other security-relevant configuration data, to ensure they are consistent with organizational security policies.

O.Time stamps

Provide time stamps to ensure that the sequencing of events can be verified.

O.User authorization management

Manage and update user authorization and privilege data to ensure they are consistent with organizational security and personnel policies.

O.React to detected attacks

Implement automated notification (or other responses) to the TSF-discovered attacks in an effort to identify attacks and to create an attack deterrent.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

5. IT Security Requirements

This section identifies the security functional and assurance requirements applicable le to the TOE. Among the security functional requirements are extended components that have been defined within the CIMC-BR-PP as indicated below.

5.1 Extended Requirements

This ST includes a number of extended requirements. Each of the extended requirements is defined in the CIMC-BR-PP and corresponding rationale immediately follows the statement of each such requirement. The extended requirements can be identified by the use of the keyword "CIMC" in the requirement component and element identifiers. These are identified in bold-italics in Table 5-1 below.

The CIMC-BR-PP should be referenced for the initial definition and rationale for each applicable requirement, though some rationale has been reproduced for each extended component included in the subsequent section as well.

5.2 TOE Security Functional Requirements

This section specifies the security requirements that are applicable to CIMC functionality, such as key management, certificate registration, and CIMC configuration and management functions.

Note that the iteration identifiers in this section are consistent with the CIMC-BR-PP and as such include reference to those CIMC-BR-PP requirements for the IT environment (as opposed to the TOE) and not necessarily considered requirements in the context of this ST. Those requirements have not been reproduced in this ST since they are not addressed by the TOE.

Table 5-1 CIMC TOE Functional Security Requirements

Security Functional Class	Security Functional Components
Security Audit (FAU)	FAU_GEN.1 Audit data generation (iteration 2)
	FAU_GEN.2 User identity association (iteration 2)
	FAU_SEL.1 Selective audit (iteration 2)
	FAU_STG.1 Protected audit trail storage (iteration 2)
	FAU_STG.4 Prevention of audit data loss (iteration 2)
Communication (FCO)	***FCO_NRO_CIMC.3 Enforced proof of origin and verification of origin***
	FCO_NRO_CIMC.4 Advanced verification of origin
Cryptographic support (FCS)	***FCS_CKM_CIMC.5 CIMC private and secret key zeroization***
	FCS_SOF_CIMC.1 CIMC Strength of Functions
User Data Protection (FDP)	FDP_ACC.1 Subset access control (iteration 2)
	FDP_ACF.1 Security attribute based access control (iteration 2)
	FDP_ACF_CIMC.2 User private key confidentiality protection
	FDP_ACF_CIMC.3 User secret key confidentiality protection
	FDP_CIMC_CER.1 Certificate Generation
	FDP_CIMC_CRL.1 Certificate Revocation
	FDP_CIMC_CSE.1 Certificate status export
	FDP_CIMC_OCSP.1 Basic Response Validation
	FDP_ETC_CIMC.5 Extended user private and secret key export
	FDP_ITT.1 Basic internal transfer protection (iterations 3 and 4)
	FDP_SDI_CIMC.3 Stored public key integrity monitoring and action
	FDP_UCT.1 Basic data exchange confidentiality (iteration 2)
Identification and authentication	FIA_SOS.1 Verification of secrets (iteration 2)

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Security Functional Class	Security Functional Components
(FIA)	FIA_UAU.1 Timing of authentication (iteration 2)
	FIA_UID.1 Timing of identification (iteration 2)
	FIA_USB.1 User-subject binding (iteration 2)
Security management (FMT)	FMT_MOF.1 Management of security functions behavior (iteration 2)
	FMT_MOF_CIMC.3 Extended certificate profile management
	FMT_MOF_CIMC.5 Extended certificate revocation list profile management
	FMT_MOF_CIMC.6 OCSP Profile Management
	FMT_MTD_CIMC.4 TSF private key confidentiality protection
	FMT_MTD_CIMC.5 TSF secret key confidentiality protection
	FMT_MTD_CIMC.7 Extended TSF private and secret key export
Protection of the TSF (FPT)	***FPT_CIMC_TSP.1 Audit log signing event***
	FPT_ITC.1 Inter-TSF confidentiality during transmission (iteration 2)
	FPT_ITT.1 Basic internal TSF data transfer protection (iterations 3 and 4)
	FPT_STM.1 Reliable time stamps (iteration 2)

5.2.1 Security Audit (FAU)

FAU_GEN.1 Audit data generation (iteration 2)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:
- a) Start-up and shutdown of the audit functions;
- b) All auditable events for the [***minimum***] level of audit; and
- c) [**the events listed in Table 5-2 below**].

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 (if applicable), 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 PP/ST, [**the information specified in the Additional Details column in Table 5-2 below**]. **Additionally, the audit shall not include plaintext private or secret keys or other critical security parameters.**

Table 5-2 Auditable Events and Audit Data

Section/Function	Component	Event	Additional Details
Security Audit	FAU_GEN.1 Audit data generation (iteration 2)	Any changes to the audit parameters, e.g., audit frequency, type of event audited Any attempt to delete the audit log	
	FPT_CIMC_TSP.1 Audit log signing event	Audit log signing event	Digital signature, keyed hash, or authentication code shall be included in the audit log.
Local Data Entry		All security-relevant data that is entered in the system	The identity of the data entry individual if the entered data is linked to any other data (e.g., clicking an "accept" button). This shall be included with the accepted data.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Section/Function	Component	Event	Additional Details
Remote Data Entry		All security-relevant messages that are received by the system	
Data Export and Output		All successful and unsuccessful requests for confidential and security-relevant information	
Key Generation	FCS_CKM.1 Cryptographic Key Generation	Whenever the TSF requests generation of a cryptographic key. (Not mandatory for single session or one-time use symmetric keys.)	The public component of any asymmetric key pair generated
Private Key Load		The loading of Component private keys	
Private Key Storage		All access to certificate subject private keys retained within the TOE for key recovery purposes	
Trusted Public Key Entry, Deletion and Storage		All changes to the trusted public keys, including additions and deletions	The public key and all information associated with the key
Secret Key Storage		The manual entry of secret keys used for authentication	
Private and Secret Key Export	FDP_ETC_CIMC.5 Extended user private and secret key export; FMT_MTD_CIMC.7 Extended TSF private and secret key export	The export of private and secret keys (keys used for a single session or message are excluded)	
Certificate Registration	FDP_CIMC_CER.1 Certificate Generation	All certificate requests	If accepted, a copy of the certificate. If rejected, the reason for rejection (e.g., invalid data, request rejected by Officer, etc.).
Certificate Status Change Approval		All requests to change the status of a certificate	Whether the request was Accepted or rejected.
CIMC Configuration		Any security-relevant changes to the configuration of the TSF.	
Certificate Profile Management	FMT_MOF_CIMC.3 Extended certificate profile management	All changes to the certificate Profile	The changes made to the Profile
Revocation Profile Management		All changes to the revocation profile	The changes made to the Profile
Certificate Revocation List Profile Management	FMT_MOF_CIMC.5 Extended certificate revocation list profile management	All changes to the certificate revocation list profile	The changes made to the profile
Online Certificate Status Protocol (OCSP) Profile Management	FMT_MOF_CIMC.6 OCSP Profile Management	All changes to the OCSP profile	The changes made to the Profile

FAU_GEN.2 User identity association (iteration 2)

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

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

FAU_SEL.1 Selective audit (iteration 2)

FAU_SEL.1.1 The TSF shall be able to select the set of audited events from the set of all auditable events based on the following attributes:
a) [***event type***]
b) [**no additional attributes**].

FAU_STG.1 Protected audit trail storage (iteration 2)

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 [***detect***] unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.4 Prevention of audit data loss (iteration 2)

FAU_STG.4.1 The TSF shall [***prevent audited events, except those taken by the ~~authorised user with special rights~~ Auditor***] and [**no other action**],if the audit trail is full.

5.2.2 Communication (FCO)

FCO_NRO_CIMC.3 Enforced proof of origin and verification of origin

FCO_NRO_CIMC.3.1 The TSF shall enforce the generation of evidence of origin for certificate status information and all other security-relevant information at all times.

FCO_NRO_CIMC.3.2 The TSF shall be able to relate the identity and [**the identity of the certificate issuer**] of the originator of the information, and the security-relevant portions of the information to which the evidence applies.

FCO_NRO_CIMC.3.3 The TSF shall verify the evidence of origin of information for all security-relevant information.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objective O.Non-repudiation and O.Control unknown source communication traffic.

FCO_NRO_CIMC.4 Advanced verification of origin

FCO_NRO_CIMC.4.1 The TSF shall, for initial certificate registration messages sent by the certificate subject, only accept messages protected using an authentication code, keyed hash, or digital signature algorithm.

FCO_NRO_CIMC.4.2 The TSF shall, for all other security-relevant information, only accept the information if it was signed using a digital signature algorithm.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objective O.Non-repudiation.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

5.2.3 Cryptographic support (FCS)

FCS_CKM_CIMC.5 CIMC private and secret key zeroization

FCS_CKM_CIMC.5.1 The TSF shall provide the capability to zeroize plaintext secret and private keys within the FIPS 140-2 validated cryptographic module.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FCS_SOF_CIMC.1 CIMC Strength of Functions

FCS_SOF_CIMC.1.1 The TSF shall provide cryptographic mechanisms that fulfill the specific Strength of Function requirements of section 10.

Rationale: This component is necessary to require specific Strength of Function metrics for cryptographic mechanisms of the TSF.

5.2.4 User Data Protection (FDP)

FDP_ACC.1 Subset access control (iteration 2)

FDP_ACC.1.1 The TSF shall enforce the [**CIMC TOE Access Control Policy specified in section 9.1]** on [**users, services, and access to services**].

FDP_ACF.1 Security attribute based access control (iteration 2)

FDP_ACF.1.1 The TSF shall enforce the [**CIMC TOE Access Control Policy specified in section 9.1]** to objects based on the following: [**identity of the subject and the set of roles that the subject is authorized to assume**].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**specified in Table 5-3**].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**no additional rules**].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [**no additional rules**].

Table 5-3 Access Controls

Section/Function	Event
Certificate Request Remote and Local Data Entry	The entry of certificate request data shall be restricted to Officers and the subject of the requested certificate.
Certificate Revocation Request Remote and Local Data Entry	The entry of certificate revocation request data shall be restricted to Officers and the subject of the certificate to be revoked.
Data Export and Output	The export or output of confidential and security-relevant data shall only be at the request of authorized users.
Key Generation	The capability to request the generation of Component keys (used to protect data in more than a single session or message) shall be restricted to Administrators.
Private Key Load	The capability to request the loading of Component private keys into cryptographic modules shall be restricted to Administrators.
Private Key	The capability to request the decryption of certificate subject private keys

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Section/Function	Event
Storage	shall be restricted to Officers. The TSF shall not provide a capability to decrypt certificate subject private keys that may be used to generate digital signatures. At least two Officers or one Officer and an Administrator, Auditor, or Operator shall be required to request the decryption of a certificate subject private key.
Trusted Public Key Entry, Deletion, and Storage	The capability to change (add, revise, delete) the trusted public keys shall be restricted to Administrators.
Secret Key Storage	The capability to request the loading of CIMC secret keys into cryptographic modules shall be restricted to Administrators.
Private and Secret Key Destruction	The capability to zeroize CIMC plaintext private and secret keys shall be restricted to Administrators, Auditors, Officers, and Operators.
Private and Secret Key Export	The capability to export a component private key shall be restricted to Administrators. The capability to export certificate subject private keys shall be restricted to Officers. The export of a certificate subject private key shall require the authorization of at least two Officers or one Officer and an Administrator, Auditor, or Operator.
Certificate Status Change Approval	Only Officers and the subject of the certificate shall be capable of requesting that a certificate be placed on hold. Only Officers shall be capable of removing a certificate from on hold status. Only Officers shall be capable of approving the placing of a certificate on hold. Only Officers and the subject of the certificate shall be capable of requesting the revocation of a certificate. Only Officers shall be capable of approving the revocation of a certificate and all information about the revocation of a certificate.

FDP_ACF_CIMC.2 User private key confidentiality protection

FDP_ACF_CIMC.2.1 CIMS personnel private keys shall be stored in a FIPS 140-2 validated cryptographic module or stored in encrypted form. If CIMS personnel private keys are stored in encrypted form, the encryption shall be performed by the FIPS 140-2 validated cryptographic module.

FDP_ACF_CIMC.2.2 If certificate subject private keys are stored in the TOE, they shall be encrypted using a Long Term Private Key Protection Key. The encryption shall be performed by the FIPS 140-2 validated cryptographic module.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FDP_ACF_CIMC.3 User secret key confidentiality protection

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

FDP_ACF_CIMC.3.1 User secret keys stored within the CIMC, but not within a FIPS 140-2 validated cryptographic module, shall be stored in encrypted form. The encryption shall be performed by the FIPS 140-2 validated cryptographic module.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FDP_CIMC_CER.1 Certificate Generation

FDP_CIMC_CER.1.1 The TSF shall only generate certificates whose format complies with [**the X.509 standard for public key certificates**].

FDP_CIMC_CER.1.2 The TSF shall only generate certificates that are consistent with the currently defined certificate profile.

FDP_CIMC_CER.1.3 The TSF shall verify that the prospective certificate subject possesses the private key that corresponds to the public key in the certificate request before issuing a certificate, unless the public/private key pair was generated by the TSF, whenever the private key may be used to generate digital signatures.

FDP_CIMC_CER.1.4 If the TSF generates X.509 public key certificates, it shall only generate certificates that comply with requirements for certificates as specified in ITU-T Recommendation X.509. At a minimum, the TSF shall ensure that:

a) The **version** field shall contain the integer **0**, **1**, or **2**.
b) If the certificate contains an **issuerUniqueID** or **subjectUniqueID** then the **version** field shall contain the integer **1** or **2**.
c) If the certificate contains **extensions** then the **version** field shall contain the integer **2**.
d) The **serialNumber** shall be unique with respect to the issuing Certification Authority.
e) The **validity** field shall specify a **notBefore** value that does not precede the current time and a **notAfter** value that does not precede the value specified in **notBefore**.
f) If the **issuer** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical **issuerAltName** extension.
g) If the **subject** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical **subjectAltName** extension.
h) The **signature** field and the **algorithm** in the **subjectPublicKeyInfo** field shall contain the OID for a FIPS-approved or recommended algorithm.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FDP_CIMC_CRL.1 Certificate revocation list validation

FDP_CIMC_CRL.1.1 A TSF that issues CRLs shall verify that all mandatory fields in any CRL issued contain values in accordance with ITU-T Recommendation X.509. At a minimum, the following items shall be validated:

1. If the **version** field is present, then it shall contain a **1**.
2. If the CRL contains any critical extensions, then the **version** field shall be present and contain the integer **1**.
3. If the **issuer** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the CRL shall contain a critical **issuerAltName** extension.
4. The **signature** and **signatureAlgorithm** fields shall contain the OID for a FIPS-approved digital signature algorithm.
5. The **thisUpdate** field shall indicate the issue date of the CRL.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

6. The time specified in the **nextUpdate** field (if populated) shall not precede the time specified in the **thisUpdate** field.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FDP_CIMC_CSE.1 Certificate status export

FDP_CIMC_CSE.1.1 Certificate status information shall be exported from the TOE in messages whose format complies with [**the X.509 standard for CRLs (RFC5280) and, the OCSP standard as defined by RFC 2560**].

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FDP_CIMC_OCSP.1 OCSP basic response validation

FDP_CIMC_OCSP.1.1 If a TSF is configured to allow OCSP responses of the basic response type, the TSF shall verify that all mandatory fields in the OCSP basic response contain values in accordance with IETF RFC 2560. At a minimum, the following items shall be validated:

1. The **version** field shall contain a **0**.
2. If the **issuer** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the response shall contain a critical **issuerAltName** extension.
3. The **signatureAlgorithm** field shall contain the OID for a FIPS-approved digital signature algorithm.
4. The **thisUpdate** field shall indicate the time at which the status being indicated is known to be correct.
5. The **producedAt** field shall indicate the time at which the OCSP responder signed the response.
6. The time specified in the **nextUpdate** field (if populated) shall not precede the time specified in the **thisUpdate** field.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FDP_ETC_CIMC.5 Extended user private and secret key export

FDP_ETC_CIMC.5.1 Private and secret keys shall only be exported from the TOE in encrypted form or using split knowledge procedures. Electronically distributed secret and private keys shall only be exported from the TOE in encrypted form.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FDP_ITT.1 Basic internal transfer protection (iteration 3)

FDP_ITT.1.1 The TSF shall enforce the [**CIMC TOE Access Control Policy specified in section 9.1]** to prevent the [***modification***] of user data when it is transmitted between physically-separated parts of the TOE.

FDP_ITT.1 Basic internal transfer protection (iteration 4)

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

FDP_ITT.1.1 The TSF shall enforce the [**CIMC TOE Access Control Policy specified in section 9.1]** to prevent the [***disclosure***] of user data when it is transmitted between physically separated parts of the TOE.

FDP_SDI_CIMC.3 Stored public key integrity monitoring and action

FDP_SDI_CIMC.3.1 Public keys stored within the CIMC, but not within a FIPS 140-2 validated cryptographic module, shall be protected against undetected modification through the use of digital signatures, keyed hashes, or authentication codes.

FDP_SDI_CIMC.3.2 The digital signature, keyed hash, or authentication code used to protect a public key shall be verified upon each access to the key. If verification fails, the TSF shall [**audit the failure**].

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FDP_UCT.1 Basic data exchange confidentiality (iteration 2)

FDP_UCT.1.1 The TSF shall enforce the [**CIMC TOE Access Control Policy specified in section 9.1]** to be able to [***transmit***] objects in a manner protected from unauthorised disclosure.

5.2.5 Identification and authentication (FIA)

FIA_SOS.1 Verification of secrets (iteration 2)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets [

1) For each attempt to use the authentication mechanism, the probability shall be less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur (e.g., guessing a password or PIN, false acceptance error rate of a biometric device, or some combination of authentication methods.) and
2) For multiple attempts to use the authentication mechanism during a one-minute period, the probability shall be less than one in 100,000 that a random attempt will succeed or a false acceptance will occur].

FIA_UAU.1 Timing of authentication (iteration 2)

FIA_UAU.1.1 The TSF shall allow [**Certificate Enrollment Requests[2], Certificate Retrieval[3] and Certificate Renewal[4]**] on behalf of the user to be performed before the user is authenticated.

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

FIA_UID.1 Timing of identification (iteration 2)

FIA_UID.1.1 The TSF shall allow [**Certificate Enrollment Requests, Certificate Retrieval, and Certificate Renewal**] on behalf of the user to be performed before the user is identified.

[2] Certificate Enrollment allows users to request various types of certificates. However, in order for a request to be fulfill the user must either be authenticated or an Officer must approve the request.
[3] Certificate Retrieval allows users to search, list and view certificates as well as download certificates and CRLs.
[4] Certificate Renewal allows users to request that their certificate be renewed. However, if they are not identified and authenticated, the request must be approved by an Officer.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

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

FIA_USB.1 User-subject binding (iteration 2)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [**authentication token**].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [**the authentication token is assigned to the user's subject after a successful authentication**].

FIA_USB.1.3 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [**the authentication token of a subject cannot change**].

5.2.6 Security management (FMT)

FMT_MOF.1 Management of security functions behavior (iteration 2)

FMT_MOF.1.1 The TSF shall restrict the ability to [***modify the behavior of***] the functions [**listed in Table 5-4**] to [**the authorized roles as specified in Table 5-4**].

Table 5-4 Authorized Roles for Management of Security Functions Behavior

Section/Function	Component Function	Authorized Role
Security Audit		The capability to configure the audit parameters shall be restricted to Administrators. The capability to change the frequency of the audit log signing event shall be restricted to Administrators.
Certificate Registration		The capability to approve fields or extensions to be included in a certificate shall be restricted to Officers. If an automated process is used to approve fields or extensions to be included in a certificate, the capability to configure that process shall be restricted to Officers.
Data Export and Output		The export of CIMC private keys shall require the authorization of at least two Administrators or one Administrator and one Officer, Auditor, or Operator.
Certificate Status Change Approval		Only Officers shall configure the automated process used to approve the revocation of a certificate or information about the revocation of a certificate. Only Officers shall configure the automated process used to approve the placing of a certificate on hold or information about the on hold status of a certificate.
CIMC		The capability to configure any TSF

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Section/Function	Component Function	Authorized Role
Configuration		functionality shall be restricted to Administrators. (This requirement applies to all configuration parameters unless the ability to configure that aspect of the TSF functionality has been assigned to a different role elsewhere in this document.)
Certificate Profile Management	FMT_MOF_CIMC.3 Extended certificate profile management	The capability to modify the certificate profile shall be restricted to Administrators.
Revocation Profile Management		The capability to modify the revocation profile shall be restricted to Administrators.
Certificate Revocation List Profile Management	FMT_MOF_CIMC.5 Extended certificate revocation list profile management	The capability to modify the certificate revocation list profile shall be restricted to Administrators.
Online Certificate Status Protocol (OCSP) Profile Management	FMT_MOF_CIMC.6 OCSP profile management	The capability to modify the OCSP profile shall be restricted to Administrators.

FMT_MOF_CIMC.3 Extended certificate profile management

FMT_MOF_CIMC.3.1 The TSF shall implement a certificate profile and shall ensure that issued certificates are consistent with that profile.

FMT_MOF_CIMC.3.2 The TSF shall require the Administrator to specify the set of acceptable values for the following fields and extensions:

- the key owner's identifier;
- the algorithm identifier for the subject's public/private key pair;
- the identifier of the certificate issuer;
- the length of time for which the certificate is valid;

FMT_MOF_CIMC.3.3 If the certificates generated are X.509 public key certificates, the TSF shall require the Administrator to specify the set of acceptable values for the following fields and extensions:

- **keyUsage**;
- **basicConstraints**;
- **certificatePolicies**

FMT_MOF_CIMC.3.4 The Administrator shall specify the acceptable set of certificate extensions.

Rationale: This component is necessary to specify a unique requirement of certificate issuing and management components that is not addressed by the CC. It supports the security objective O.Configuration management.

FMT_MOF_CIMC.5 Extended certificate revocation list profile management

FMT_MOF_CIMC.5.1 If the TSF issues CRLs, the TSF must implement a certificate revocation list profile and ensure that issued CRLs are consistent with the certificate revocation list profile.

FMT_MOF_CIMC.5.2 If the TSF issues CRLs, the TSF shall require the Administrator to specify the set of acceptable values for the following fields and extensions:

- **issuer;**

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

- **issuerAltName** (NOTE: If a CIMC does not issue CRLs with this extension, then it is not required within the certificate revocation list profile.)
- **nextUpdate** (i.e., a promise of next CRL in specified time).

FMT_MOF_CIMC.5.3 If the TSF issues CRLs, the Administrator shall specify the acceptable set of CRL and CRL entry extensions.

Rationale: This component is necessary to specify a unique requirement of certificate issuing and management components that is not addressed by the CC. It supports the security objective O.Configuration management.

FMT_MOF_CIMC.6 OCSP profile management

FMT_MOF_CIMC.6.1 If the TSF issues OCSP responses, the TSF shall implement an OCSP profile and ensure that issued OCSP responses are consistent with the OCSP profile.

FMT_MOF_CIMC.6.2 If the TSF issues OCSP responses, the TSF shall require the Administrator to specify the set of acceptable values for the **responseType** field (unless the CIMC can only issue responses of the basic response type).

FMT_MOF_CIMC.6.3 If the TSF is configured to allow OCSP responses of the basic response type, the TSF shall require the Administrator to specify the set of acceptable values for the **ResponderID** field within the basic response type.

Rationale: This component is necessary to specify a unique requirement of certificate issuing and management components that is not addressed by the CC. It supports the security objective O.Configuration management.

FMT_MTD_CIMC.4 TSF private key confidentiality protection

FMT_MTD_CIMC.4.1 CIMC private keys shall be stored in a FIPS 140-2 validated cryptographic module or stored in encrypted form. If CIMC private keys are stored in encrypted form, the encryption shall be performed by the FIPS 140-2 validated cryptographic module.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FMT_MTD_CIMC.5 TSF secret key confidentiality protection

FMT_MTD_CIMC.5.1 TSF secret keys stored within the TOE, but not within a FIPS 140-2 validated cryptographic module, shall be stored in encrypted form. The encryption shall be performed by the FIPS 140-2 validated cryptographic module.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

FMT_MTD_CIMC.7 Extended TSF private and secret key export

FMT_MTD_CIMC.7.1 Private and secret keys shall only be exported from the TOE in encrypted form or using split knowledge procedures. Electronically distributed secret and private keys shall only be exported from the TOE in encrypted form.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by the CC.

5.2.7 Protection of the TSF (FPT)

FPT_CIMC_TSP.1 Audit log signing event

FPT_CIMC_TSP.1.1 The TSF shall periodically create an audit log signing event in which it computes a digital signature, keyed hash, or authentication code over the entries in the audit log.

FPT_CIMC_TSP.1.2 The digital signature, keyed hash, or authentication code shall be computed over, at least, every entry that has been added to the audit log since the previous audit log signing event and the digital signature, keyed hash, or authentication code from the previous audit log signed event.

FPT_CIMC_TSP.1.3 The specified frequency at which the audit log signing event occurs shall be configurable.

FPT_CIMC_TSP.1.4 The digital signature, keyed hash, or authentication code from the audit log signing event shall be included in the audit log.

Rationale: This component is necessary to specify a unique requirement for certificate issuing and management components that is not addressed by existing CC requirements. It supports the security objective O.Protect stored audit records, by providing additional protection for stored audit records..

FPT_ITC.1 Inter-TSF confidentiality during transmission (iteration 2)

FPT_ITC.1.1 The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorised disclosure during transmission.

FPT_ITT.1 Basic internal TSF data transfer protection (iteration 3)

FPT_ITT.1.1 The TSF shall protect TSF data from [***modification***] when it is transmitted between separate parts of the TOE.

FPT_ITT.1 Basic internal TSF data transfer protection (iteration 4)

FPT_ITT.1.1 The TSF shall protect TSF data from [***disclosure***] when it is transmitted between separate parts of the TOE.

FPT_STM.1 Reliable time stamps (iteration 2)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 (EAL 4) components, as specified in Part 3 of CCv3.1, augmented with ALC_FLR.2 as indicated in bold the following table.

Table 5-5 Assurance Requirements (EAL 4 augmented)

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.4: Complete functional specification
	ADV_IMP.1: Implementation representation of the TSF
	ADV_TDS.3: Basic modular design

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Requirement Class	Requirement Component
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.4: Production support, acceptance procedures and automation
	ALC_CMS.4: Problem tracking CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_DVS.1: Identification of security measures
	ALC_FLR.2: Flaw reporting procedures
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.2: Testing: security enforcing modules
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.3: Focused vulnerability analysis

5.3.1 Development (ADV)

ADV_ARC.1 Security architecture description

ADV_ARC.1.1d The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.
ADV_ARC.1.2d The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.
ADV_ARC.1.3d The developer shall provide a security architecture description of the TSF.
ADV_ARC.1.1c The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.
ADV_ARC.1.2c The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.
ADV_ARC.1.3c The security architecture description shall describe how the TSF initialisation process is secure.
ADV_ARC.1.4c The security architecture description shall demonstrate that the TSF protects itself from tampering.
ADV_ARC.1.5c The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
ADV_ARC.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.4 Complete functional specification

ADV_FSP.4.1d The developer shall provide a functional specification.
ADV_FSP.4.2d The developer shall provide a tracing from the functional specification to the SFRs.
ADV_FSP.4.1c The functional specification shall completely represent the TSF.
ADV_FSP.4.2c The functional specification shall describe the purpose and method of use for all TSFI.
ADV_FSP.4.3c The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.4.4c The functional specification shall describe all actions associated with each TSFI.
ADV_FSP.4.5c The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.
ADV_FSP.4.6c The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.
ADV_FSP.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.4.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

ADV_IMP.1 Implementation representation of the TSF

ADV_IMP.1.1d The developer shall make available the implementation representation for the entire TSF.
ADV_IMP.1.2d The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.
ADV_IMP.1.1c The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.1.2c The implementation representation shall be in the form used by the development personnel.
ADV_IMP.1.3c The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.
ADV_IMP.1.1e The evaluator shall confirm that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.3 Basic modular design

ADV_TDS.3.1d The developer shall provide the design of the TOE.
ADV_TDS.3.2d The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.
ADV_TDS.3.1c The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.3.2c The design shall describe the TSF in terms of modules.
ADV_TDS.3.3c The design shall identify all subsystems of the TSF.
ADV_TDS.3.4c The design shall provide a description of each subsystem of the TSF.
ADV_TDS.3.5c The design shall provide a description of the interactions among all subsystems of the TSF.
ADV_TDS.3.6c The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.
ADV_TDS.3.7c The design shall describe each SFR-enforcing module in terms of its purpose and interaction with other modules.
ADV_TDS.3.8c The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, interaction with and called interfaces to other modules.
ADV_TDS.3.9c The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.
ADV_TDS.3.10c The mapping shall demonstrate that all behaviour described in the TOE design is mapped to the TSFIs that invoke it.
ADV_TDS.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_TDS.3.2e The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance documents (AGD)

AGD_OPE.1 Operational user guidance

AGD_OPE.1.1d The developer shall provide operational user guidance.
AGD_OPE.1.1c The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
AGD_OPE.1.2c The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.
AGD_OPE.1.3c The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.
AGD_OPE.1.4c The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

AGD_OPE.1.5c The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AGD_OPE.1.6c The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.
AGD_OPE.1.7c The operational user guidance shall be clear and reasonable.
AGD_OPE.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1 Preparative procedures

AGD_PRE.1.1d The developer shall provide the TOE including its preparative procedures.
AGD_PRE.1.1c The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2c The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.
AGD_PRE.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2e The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.3.3 Life-cycle support (ALC)

ALC_CMC.4 Production support, acceptance procedures and automation

ALC_CMC.4.1d The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.4.2d The developer shall provide the CM documentation.
ALC_CMC.4.3d The developer shall use a CM system.
ALC_CMC.4.1c The TOE shall be labelled with its unique reference.
ALC_CMC.4.2c The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.4.3c The CM system shall uniquely identify all configuration items.
ALC_CMC.4.4c The CM system shall provide automated measures such that only authorised changes are made to the configuration items.
ALC_CMC.4.5c The CM system shall support the production of the TOE by automated means.
ALC_CMC.4.6c The CM documentation shall include a CM plan.
ALC_CMC.4.7c The CM plan shall describe how the CM system is used for the development of the TOE.
ALC_CMC.4.8c The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.
ALC_CMC.4.9c The evidence shall demonstrate that all configuration items are being maintained under the CM system.
ALC_CMC.4.10c The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.
ALC_CMC.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_CMS.4 Problem tracking CM coverage

ALC_CMS.4.1d The developer shall provide a configuration list for the TOE.
ALC_CMS.4.1c The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.
ALC_CMS.4.2c The configuration list shall uniquely identify the configuration items.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

ALC_CMS.4.3c For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.
ALC_CMS.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DEL.1 Delivery procedures

ALC_DEL.1.1d The developer shall document procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2d The developer shall use the delivery procedures.
ALC_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.
ALC_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1 Identification of security measures

ALC_DVS.1.1d The developer shall produce development security documentation.
ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.
ALC_DVS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

ALC_FLR.2 Flaw reporting procedures

ALC_FLR.2.1d The developer shall document flaw remediation procedures addressed to TOE developers.
ALC_FLR.2.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.
ALC_FLR.2.3d The developer shall provide flaw remediation guidance addressed to TOE users.
ALC_FLR.2.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.
ALC_FLR.2.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.
ALC_FLR.2.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.
ALC_FLR.2.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.
ALC_FLR.2.5c The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.
ALC_FLR.2.6c The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.
ALC_FLR.2.7c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.
ALC_FLR.2.8c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.
ALC_FLR.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_LCD.1 Developer defined life-cycle model

ALC_LCD.1.1d The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.
ALC_LCD.1.2d The developer shall provide life-cycle definition documentation.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

ALC_LCD.1.1c The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.
ALC_LCD.1.2c The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.
ALC_LCD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_TAT.1 Well-defined development tools

ALC_TAT.1.1d The developer shall identify each development tool being used for the TOE.
ALC_TAT.1.2d The developer shall document the selected implementation-dependent options of each development tool.
ALC_TAT.1.1c Each development tool used for implementation shall be well-defined.
ALC_TAT.1.2c The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.
ALC_TAT.1.3c The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.
ALC_TAT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Tests (ATE)

ATE_COV.2 Analysis of coverage

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.
ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.
ATE_COV.2.2c The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.
ATE_COV.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_DPT.2 Testing: security enforcing modules

ATE_DPT.2.1d The developer shall provide the analysis of the depth of testing.
ATE_DPT.2.1c The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems and SFR-enforcing modules in the TOE design.
ATE_DPT.2.2c The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.
ATE_DPT.2.3c The analysis of the depth of testing shall demonstrate that the SFR-enforcing modules in the TOE design have been tested.
ATE_DPT.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 Functional testing

ATE_FUN.1.1d The developer shall test the TSF and document the results.
ATE_FUN.1.2d The developer shall provide test documentation.
ATE_FUN.1.1c The test documentation shall consist of test plans, expected test results and actual test results.
ATE_FUN.1.2c The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.3c The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.4c The actual test results shall be consistent with the expected test results.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 Independent testing – sample

ATE_IND.2.1d The developer shall provide the TOE for testing.
ATE_IND.2.1c The TOE shall be suitable for testing.
ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.
ATE_IND.2.3e The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.5 Vulnerability assessment (AVA)

AVA_VAN.3 Focused vulnerability analysis

AVA_VAN.3.1d The developer shall provide the TOE for testing.
AVA_VAN.3.1c The TOE shall be suitable for testing.
AVA_VAN.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VAN.3.2e The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.
AVA_VAN.3.3e The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.
AVA_VAN.3.4e The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Enhanced-Basic attack potential.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Identification & Authentication

Users are identified using certificates. The certificates are originally verified by the HTTP Engine using SSL. The certificates are passed to RHCS 8.1 which attempts to match the certificate with a user in its internal database. If this is successful, an authentication token is created with the user attributes associated with the certificate (e.g., user roles). Subsequent requests from the same SSL session are associated with this authentication token.

If a certificate is invalid, a failure will occur in establishing the SSL session and no services will be available. If a certificate is valid, but not recognized by RHCS 8.1, identification and authentication will fail and applicable services will not be available.

The use of certificates for authentication will ensure that the authentication data always has acceptable strength. Specifically, the probability of a user successfully forging a certificate would by much less than 1 in 1,000.000 and within a minute it would be impossible to reduce the probably to less than 1 in 100,000 given the strength of the certificates and the limitations on making attempts via accessible interfaces.

There are a number of services that are available from the TOE that do not require authentication. These services include:

- Enrollment Requests
  - A user can request various types of certificates (note that the request will not be granted unless the user is authenticated or an Agent approves the request)
  - A user can also request to renew a certificate, but just like enrollment, the user must either be authenticated (for automatic issuance) or the request must be approved by an Agent
- Retrieval
  - Check request status
  - List certificates
  - Search certificates
  - Import CA certificate chain
  - Import certificate revocation list

Note that only these services are available without using SSL with client authentication. When using SSL with client authentication, additional features are provided. These features are Renewal and Revocation services, which require client certificates for authentication. Authentication is required for these services to ensure that users can renew or revoke only their own certificates.

Agent functions are available using SSL client authentication. Other administrative functions are available using the Console application; except for the TPS where applicable functions are either performed in the IT environment by modifying configuration files (under the control of the host operating system) or by using SSL client authentication via a browser application. The Console application uses SSL where client authentication is enforced in the TOE configuration. Note that all of the communication ports are configured by default and can be changed by an Administrator.

The Identification & Authentication security function satisfies the following security requirements:

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

FIA_SOS.1 (iteration 2) – RHCS 8.1 requires certificate based authentication that has a strength much greater than that required even when guessing at the maximum rate possible using the TOE interfaces..

FIA_UAU.1 (iteration 2) – RHCS 8.1 only allows enrollment requests and certificate related retrieval requests without being authenticated. The identification and authentication requires a valid certificate, known to RHCS 8.1.

FIA_UID.1 (iteration 2) – RHCS 8.1 only allows enrollment requests and certificate related retrieval requests without being identified. The identification and authentication requires a valid certificate, known to RHCS 8.1.

FIA_USB.1 (iteration 2) – RHCS 8.1 ensures that users are associated with their actions by creating an authentication token when a user is identified and authenticated, and then associating that authentication token with every request made in the context of the corresponding SSL session.

6.1.2 Access Control

With the exception of the TPS component, each servlet (i.e., service) has an access control list that defines which users and groups can use the services of that servlet. These access control lists simply list the users and/or groups that are permitted to invoke the servlet. When a request comes in to access a servlet, the user (and associated groups) is checked against the access control list on the servlet and the servlet will execute only if the user is allowed.

In the case of TPS, access is fixed for each role rather than having a configurable access control list. When a request comes in to access the TPS servlet, whether the request will succeed is dependent upon the hard-coded function restrictions based on the authenticated user and associated groups (i.e., role).

Users can access the TOE only using the HTTP-based interfaces (including the console application). The only accesses not subject to access control are those accessible outside a SSL session (i.e., those that do not require identification or authentication - see above). By enforcing an access control check on all other accesses, RHCS 8.1 ensures that its access control mechanism cannot be bypassed.

The Access Control security function satisfies the following security requirements:

FDP_ACC.1 (iteration 2) – RHCS 8.1 includes a number of services and each is assigned an access control list (or is hard-coded) defining who can access the service. Users are defined internally in RHCS 8.1 and once authenticated, their user identity and associated roles are used to make access decisions.

FDP_ACF.1 (iteration 2) – RHCS 8.1 uses its access control mechanisms primarily to enforce user access and role restrictions define in Table 6-1. Note that there are some operations where the subject of the certificate is allowed to request an operation on the certificate – in these cases a Proof of Possession (POP) check is performed to ensure the certificate belongs to the requesting subject.

6.1.3 Security Management

RHCS 8.1 can be configured to define specific groups (or roles). Each group can be assigned one or more users. The Access Control mechanism is used to restrict functions to specific administrator roles by configuring necessary access control lists.

6.1.3.1 RHCS 8.1 Privileged Users and Groups (Roles)

Each Certificate System (CS) subsystem has up to five roles. The roles that are created are specific to the CS subsystem, and depend on which CS subsystem has been installed. All of the privileged roles (see About Roles for more information about privileges) require SSL client-authentication by presenting a certificate that maps to the user with the corresponding role (i.e., authorization). The following sections show the default roles that are created with each subsystem and the main privileges of each.
Note that “Trusted Manager” is a logical role where referenced and is a notion used to define accounts used for the purpose of communication among the associated subsystems (i.e., a security domain). Additionally, each subsystem

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

has the concept of an “Enterprise Administrator” that is responsible to install the subsystem and perform basic configuration functions associated with the association of the subsystem with other subsystems (e.g., adding to or removing from a given security domain), but that role has no additional responsibility (i.e., it is not a superset of the “Administrator” role for example).

6.1.3.1.1 CA

- Enterprise Administrator
    - Can run installation servlets (e.g., to get configuration information, register agents, import certificates, modify OCSP and connector information).
    - Can modify the security domain (i.e., association with other subsystems associated with the CA).
- Administrators
    - Can start/stop the server (from the command-line).
    - Can perform all configuration management for CA (unless assigned otherwise), including the configuration of certificate profiles (specifying the set of acceptable values for fields and extensions) for certificate enrollment requests (via the CS Console).
- Certificate Manager Agents
    - Can approve fields/extensions (to be included in a certificate) of certificate profiles that have been enabled and configured by the Administrator (via SSL-capable browsers to the CA Agent interface).
    - Can run tools (CMCEnroll and CMCRevoke) to pre-approve certificate enrollment and revocation requests.
- Auditors
    - Can view signed audit logs (from the IT environment). This is the only role allowed this privilege.
    - Can verify audit log signatures by running the AuditVerify tool.
- Trusted Manager
    - The Trusted Manager role is a special role that is not for privileged users. It is created for inter-CIMC_boundary communication. The trust of this communication is established using the role authentication/authorization mechanism. Conceptually, this role is not an actual privileged role that a user can be assigned to. Rather, the Trusted Manager role is a means of establishing trust between two CS subsystems. To have the TPS communicate with the CA securely, a "TPS user" is created on the CA with the Trusted Manager role during setup. All communications between the TPS and CA are then made through this special user with the TPS's certificate over SSL client-authentication and the Trusted Manager role authorization (via Inter-CIMC_boundary interface connectors).

6.1.3.1.2 DRM

- Enterprise Administrator

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

  - Can run installation servlets (e.g., to get configuration information, register agents, import certificates, modify OCSP and connector information).

- Administrators
  - Can start/stop server (from the command-line).
  - Can perform all configuration management for the DRM (via the CS Console).
- Data Recovery Manager Agents
  - Can approve recovery of subject private keys (via SSL-capable browsers to the DRM Agent interface).
  - Can export recovered subject private keys (via SSL-capable browsers to the DRM Agent interface).
- Auditors
  - Can view signed audit logs (from the IT environment). This is only role allowed this privilege.
  - Can verify audit log signatures by running the AuditVerify tool.
- Trusted Manager
  - The Trusted Manager role is a special role that is not for privileged users. It is created for inter-CIMC_boundary communication. The trust of this communication is established using the role authentication/authorization mechanism. Conceptually, this role is not an actual privileged role that a user can be assigned to. Rather, the Trusted Manager role is a means of establishing trust between two CS subsystems. To have the CA and TPS communicate with the DRM securely[5], CA and TPS users are created on the DRM with the Trusted Manager role during setup. All communications between the CA/TPS and DRM are then made through this special user with the CA/TPS's certificate over SSL client-authentication and Trusted Manager role authorization.

6.1.3.1.3 OCSP

- Enterprise Administrator
  - Can run installation servlets (e.g., to get configuration information, register agents, import certificates, modify OCSP and connector information).
- Administrators
  - Can start/stop server (from the command-line).
  - Can perform all configuration management for DRM (via the CS Console).
- Online Certificate Status Manager Agents
  - Can add CRLs (to the OCSP Responder Agent interface via SSL-capable browsers).
  - Can define supported CAs (via SSL-capable browsers to the OCSP Responder Agent interface).

[5] Note that the TOE can be configured as a Token Management System (TMS) or a non-TMS. When configured as a TMS, the CA does not communicate directly with the DRM, but rather communication is handled indirectly via the TPS. In a non-TMS configuration the CA can be configured to directly communicate with the DRM.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

- Auditors
    - Can view signed audit logs (via the CS Console). This is the only role allowed this privilege.
    - Can verify audit log signatures by running the AuditVerify tool.

6.1.3.1.4 TKS

- Enterprise Administrator
    - Can run installation servlets (e.g., to get configuration information, register agents, import certificates, modify OCSP and connector information).
- Administrators
    - Can start/stop the server (from the command-line).
    - Can perform all configuration management for TKS (unless assigned otherwise), including the configuration of token policies (via a combination of configuration file manipulation and use of the CS Console).
- Auditors
    - Can view signed audit logs (from the IT environment). This is the only role allowed this privilege.
    - Can verify audit log signatures by running the AuditVerify tool.
- Trusted Manager
    - The Trusted Manager role is a special role that is not for privileged users. It is created for inter-CIMC_boundary communication. The trust of this communication is established using the role authentication/authorization mechanism. Conceptually, this role is not an actual privileged role that a user can be assigned to. Rather, the Trusted Manager role is a means of establishing trust between two CS subsystems. To have the TPS communicate with the TKS securely, a "TPS user" is created on the TKS with the Trusted Manager role during setup. All communications between the TPS and TKS are then made through this special user with the TPS's certificate over SSL client-authentication and the Trusted Manager role authorization (via Inter-CIMC_boundary interface connectors).

6.1.3.1.5 TPS

- Enterprise Administrator
    - Can run installation servlets (e.g., to get configuration information, register agents, import certificates, modify OCSP and connector information).
- Administrators
    - Can start/stop the server (from the command-line).
    - Can perform all configuration management for TPS (unless assigned otherwise), including user and token (add or delete, but not edit), audit log configuration, and TPS activity monitoring

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

(search and list) (via a combination of configuration file manipulation or SSL-capable browsers to the admin tab on the TPS Agent interface).

- Token Processing System Agents
    - Can manage tokens (but not add or delete them); edit policy, status, and user ID bindings; and list and search certificates. Can also view activities associated with those tokens.
- Operators
    - Can examine tokens, policies, status, user ID bindings, certificates, and TPS activities (associated with those tokens).
- Auditors
    - Can view signed audit logs (from the IT environment). This is the only role allowed this privilege.
    - Can verify audit log signatures by running the AuditVerify tool.

6.1.3.2 About Roles

Of all privileged roles supported by the CS, the Certificate Manager Agents role, the Data Recovery Manager Agents role, and the Token Processing System Agent Role are the ones that map directly to the "Officer" role defined in the ST and the CIMC-BR-PP. The Online Certificate Status Manager Agents are a sub-group of the Administrator role defined in the CIMC-BR-PP. The TPS Operator role is a sub-group of the TPS agent role, but doesn't have any 'write' access. The following further specifies this mapping:

- Administrator

  The Administrator role is divided into finer-grained sub-roles, each bearing different responsibilities:
    - Enterprise Administrators for the CA, DRM, OCSP, TKS, and TPS subsystems
    - Administrators for the CA, DRM, OCSP, TKS, and TPS subsystems
    - Online Certificate Status Manager Agents
- Officer
    - Certificate Manager Agents
    - Data Recovery Manager Agents
    - Token Processing System Agents
- Auditor
    - Auditors from CA, DRM, OCSP, TKS, and TPS

6.1.3.3 Access Rules:

The following access rules are used to establish the default access control lists for the servlets. Note that the access control lists used only to restrict functions associated with explicitly defined users and groups (i.e., roles). Rules restricting access to subjects of certificates are enforced directly using certificate-based identification and authentication or POP.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Table 6-1 Role Restrictions

Section/Function	Authorized Role
Required by FDP_ACF	
Certificate Request Remote and Local Data Entry	The entry of certificate request data is restricted to Officers *and the subject of the requested certificate*.
Certificate Revocation Request Remote and Local Data Entry	The entry of certificate revocation request data is restricted to Officers *and the subject of the certificate to be revoked*.
Data Export and Output	The export or output of confidential and security-relevant data is performed only at the request of authorized users.
Key Generation	The capability to request the generation of Component keys (used to protect data in more than a single session or message) is restricted to Administrators.
Private Key Load	The capability to request the loading of Component private keys into cryptographic modules is restricted to Administrators.
Private Key Storage	The capability to request the decryption of certificate subject private keys is restricted to Officers. RHCS 8.1 does not provide a capability to decrypt certificate subject private keys that may be used to generate digital signatures. RHCS 8.1 does not allow users or administrators to decrypt a certificate subject private key.
Trusted Public Key Entry, Deletion, and Storage	The capability to change (add, revise, delete) the trusted public keys is restricted to Administrators.
Secret Key Storage	The capability to request the loading of CIMC secret keys into cryptographic modules is restricted to Administrators.
Private and Secret Key Destruction	The capability to delete CIMC plaintext private and secret keys is not available to any role. Key destruction, implemented by zeroization, is provided by the underlying FIPS compliant HSM where such functionality is called by NSS when necessary.
Private and Secret Key Export	The capability to export a component private key is restricted to Administrators. The capability to export certificate subject private keys is restricted to Officers. RHCS 8.1 does not support the export of component private keys. The export of a certificate subject private key requires the authorization of at least two Officers.
Certificate Status Change Approval	Only Officers *and the subject of the certificate* are capable of requesting that a certificate be placed on hold. Only Officers are capable of removing a certificate from on hold status. Only Officers are capable of approving the placing of a certificate on hold. Only Officers *and the subject of the certificate* are capable of requesting the revocation of a certificate.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Section/Function	Authorized Role
	Only Officers are capable of approving the revocation of a certificate and all information about the revocation of a certificate.
Required by FMT_MOF	
Security Audit	The capability to configure the audit parameters is restricted to Administrators. The capability to change the frequency of the audit log signing event is restricted to Administrators.
Certificate Registration	The capability to approve fields or extensions to be included in a certificate is restricted to Officers. If an automated process is used to approve fields or extensions to be included in a certificate, the capability to configure that process shall be restricted to Officers.
Data Export and Output	RHCS 8.1 does not support the export of CIMC private keys.
Certificate Status Change Approval	Only Officers can configure the automated process used to approve the revocation of a certificate or information about the revocation of a certificate. Only Officers can configure the automated process used to approve the placing of a certificate on hold or information about the on hold status of a certificate.
CIMC Configuration	Except as stated elsewhere, the capability to configure any TSF functionality is restricted to Administrators.
Certificate Profile Management	The capability to modify the certificate profile is restricted to Administrators.
Revocation Profile Management	The capability to modify the revocation profile is restricted to Administrators.
Certificate Revocation List Profile Management	The capability to modify the certificate revocation list profile is restricted to Administrators.
Online Certificate Status Protocol (OCSP) Profile Management	The capability to modify the OCSP profile is restricted to Administrators.

The Security Management security function satisfies the following security requirements:

> FMT_MOF.1 (iteration 2) – RHCS 8.1 uses the access control mechanism to ensure that the various security roles can only perform appropriate functions as indicated in the table above.

6.1.4 Security Audit

RHCS 8.1 maintains all security relevant audit records in an audit log. The audit log is managed by a logging component that is used by the CA, OCSP Responder, DRM, TKS, or TPS (which has its own audit subsystem) whenever an event occurs that requires logging.

Each audit record includes:

- date,
- time,
- event type,
- thread ID,
- responsible user or agent,

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

- indication of success or failure,
- and other relevant information depending on the event type:
    - request identifier,
    - authentication source,
    - state,
    - DN,
    - Serial number,
    - Violation indicator,
    - Reason indicator

The following table lists the minimum set of auditable events (and additional audit record details when applicable):

Table 6-2 Auditable Events

Event	Additional Details
Changes to the audit parameters	
Attempts to delete the audit log	
Startup and shutdown of the audit function	
Audit log signing event	Digital signature
Modifications to the audit configuration (while the audit collection functions are operating)	
Successful requests to perform an operation on an object covered by the SFP	
Successful transfers of user data	Identification of the protection method used
The identity of any user or subject using the data exchange mechanisms	
Unsuccessful use of the user identification and authentication mechanism, including the user identity provided	
Unsuccessful binding of user security attributes to a subject (e.g. creation of a subject)	
Changes to the time	
All security-relevant data that is entered in the system	The identity of the data entry individual if the entered data is linked to any other data (e.g., clicking an “accept” button). This shall be included with the accepted data.
All security-relevant messages (i.e., requests) that are received by the system	
Successful and unsuccessful requests for confidential and security-relevant information	
Whenever the TSF requests generation of a cryptographic key. (Not mandatory for single session or one-time use symmetric keys.)	The public component of any asymmetric key pair generated
The loading of Component private keys	
Access to certificate subject private keys retained within the TOE for key recovery purposes	
Changes to the trusted public keys, including additions and deletions	The public key and all information associated with the key
Manual entry of secret keys used for authentication	
Export of private and secret keys (keys used for a single session or message are excluded)	
Certificate requests	If accepted, a copy of the certificate. If rejected, the reason for rejection (e.g., invalid data, request rejected by Officer, etc.).
Requests to change the status of a certificate	Whether the request was accepted or rejected.
Security-relevant changes to the configuration of the TSF.	

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Event	Additional Details
All changes to the certificate Profile	The changes made to the Profile
Changes to the revocation profile	The changes made to the Profile
Changes to the certificate revocation list profile	The changes made to the profile
Changes to the OCSP profile	The changes made to the Profile

The logging subsystem can filter audit records as they occur. The following attributes can be selected to be excluded from the audit log:

- event type

Note that all audit records are included by default and the selection rules can be used to reduce the set of audit records that are included in the audit log.

When a write to the audit log fails, RHCS 8.1 will shutdown to prevent additional auditable events to be generated. Subsequently, the TOE will not start until the situation is resolved by collaborative effort of the Auditor and Administrator in the IT Environment.

The audit log is stored internal to the RHCS 8.1 system, albeit using file constructs made available by the host operating system. The only interfaces offered to delete audit records are controlled using an access control list so that no user can delete audit records or an entire audit log through the CS TOE. Removal of an audit log must be done through the IT environment by a CS auditor. In order to prevent undetected modification of audit records, RHCS 8.1 can be configured to sign entries in the log. Each signature is itself written as an entry in the audit log after the entries that are signed. The signature is computed over the previous log entries, starting with, and including, the previous signature. Since the previous signature is signed along with the intervening data, the signatures form a chain reaching back to the very first signature created by the CS instance. This chaining property can be used to detect the insertion of bogus log entries before a block of signed log entries, or the deletion of a block of log entries. The interval of flushing audit buffers (and the signing of which) to a file is configurable by the Administrator in RHCS 8.1.

RHCS 8.1 ensures that each record includes a reliable time stamp by always obtaining the current time and date from its host.

The Security Audit security function satisfies the following security requirements:

FAU_GEN.1 (iteration 2) – RHCS 8.1 minimally generates the events listed in the table above and includes the date, time, event type, subject, success or failure, as well as any additional content listed in the table above.

FAU_GEN.2 (iteration 2) – RHCS 8.1 records the responsible user in the contents of each audit record. The user identity is the target user for failed authentication attempts or the user authenticated for the session causing the event.

FAU_SEL.1 (iteration 2) – RHCS 8.1 includes the ability to filter audit records based on their event type as they occur.

FAU_STG.1 (iteration 2) – RHCS 8.1 protects audit records using access controls that allow only an Auditor to review or delete the audit log. RHCS 8.1 provides additional assurance that audit records are not modified by digitally signing audit record buffers as they are flushed into the non-volatile audit log storage.

FAU_STG.4 (iteration 2) – When the audit log becomes full, RHCS 8.1 shuts down and will not start until the condition is addressed.

FPT_STM.1 (iteration 2) – RHCS 8.1 ensures that reliable time stamps are included with each audit record by always obtaining the current time from its host.

FPT_CIMC_TSP.1 – RHCS 8.1 signs each audit buffer as it is flushed to non-volatile storage. The signature includes the keyed hash of the previous buffer to ensure a whole buffer cannot be removed, and each signature is stored along with its buffer.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

6.1.5 Remote Data Entry & Export

RHCS 8.1 is responsible for importing and exporting certificates, keys, key components, certificate status, and other data. RHCS 8.1 protects these data transfers from unauthorized disclosure and modification using SSL sessions or CRMF/PKCS#10 signatures in the case of certificate requests. In addition, the TOE provides certificate status information by following means: OCSP messages and CRLs.

The Remote Data Entry & Export security function satisfies the following security requirements:

FCO_NRO_CIMC.3 – RHCS 8.1 generates digital signatures for certificates, CRLs, and OCSPs. Inbound requests are authenticated using SSL or CRMF/PKCS#10 in the case of certificate requests.

FDP_UCT.1 (iteration 2) – All communications external to the TOE and internal on remote components are performed over a SSL session. The SSL session will protect the data transmitted from unauthorized modification or disclosure.

FPT_ITC.1 (iteration 2) – All communications external to the TOE and internal on remote components are performed over a SSL session. The SSL session will protect the data transmitted from unauthorized modification or disclosure.

FCO_NRO_CIMC.4 - CRMF/PKCS#10 signatures are used to verify certificate requests, all other security relevant messages are verified using SSL.

FDP_CIMC_CSE.1 - The TOE provides certificate status information by following means:

1. OCSP messages (RFC 2560 compliant)
2. CRLs (X.509 / RFC 5280 compliant)

FDP_ITT.1 (iteration 3 & 4) - All communications external to the TOE and internal on remote components are performed over a SSL session. The SSL session will protect the data transmitted from unauthorized modification or disclosure.

FPT_ITT.1 (iteration 3 & 4) - All communications external to the TOE and internal on remote components are performed over a SSL session. The SSL session will protect the data transmitted from unauthorized modification or disclosure.

6.1.6 Key Management

RHCS 8.1 supports key generation for certificates and encryption and import and export of public and private keys. RHCS 8.1 relies on a FIPS 140-2 validated module to perform critical key generation, key storage, and zeroization for key destruction. Additional details can be found in the security requirement mapping below.

The Key Management security function satisfies the following security requirements:

FDP_ACF_CIMC.2 – RHCS 8.1 does not support CIMS personnel private keys. Certificate private keys are encrypted using a hardware cryptographic module, but are not stored within the TOE.

FMT_MTD.CIMC.4 – RHCS 8.1 stores all CIMC private keys in a hardware cryptographic module.

FDP_SDI_CIMC.3 – Public keys are all stored signed with a digital signature. The signature on the digital certificate is verified each time the key is accessed. If the verification fails an audit record is generated and the certificate cannot be used.

FDP_ACF_CIMC.3 – RHCS 8.1 does not store user secret keys.

FMT_MTD_CIMC.5 – RHCS 8.1 secret keys are stored in hardware cryptographic modules.

FCS_CKM_CIMC.5 – RHCS 8.1 does not store plaintext keys itself, but does invoke zeroization functions provided by the hardware cryptographic modules.

FDP_ETC_CIMC.5 – RHCS 8.1 only exports private/secret keys for DRM private key restoration. This export is always in encrypted form.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

FMT_MTD_CIMC.7 – RHCS 8.1 does not export TSF private or secret keys.

The CS uses the PKCS# module provided by the cryptographic hardware venders to access the hardware cryptographic modules. These cryptographic hardware components are expected to have been successfully evaluated through the FIPS 140-2 program.

6.1.7 Certificate Management

RHCS 8.1 provides functionality to issue, suspend, reinstate, renew, and revoke certificates, report status of certificates, and generate CRLs and OCSP responses. All these certificate services are provided in a secure manner, protecting the integrity of the certificates. Additionally, RHCS 8.1 enforces proof of possession to ensure that certificates are issued securely. RHCS 8.1 offers administrators the ability to configure profiles that are applied to certificates. These profiles either remove disallowed content or add mandatory content as certificate requests are processed. The security requirement mapping below describes minimum capabilities provided by RHCS 8.1.

The Certificate Management security function satisfies the following security requirements:

FMT_MOF_CIMC.3 – RHCS 8.1 requires the Administrator to specify the set of acceptable values for:

- the key owner's identifier;
- the algorithm identifier for the subject's public/private key pair;
- the identifier of the certificate issuer;
- the length of time for which the certificate is valid.;
- **keyUsage**;
- **basicConstraints**;
- **certificatePolicies;** and
- acceptable certificate extensions.

FMT_MOF_CIMC.5 – RHCS 8.1 allows the Administrator to define a CRL profile that constrains CRLs. The Administrator must specify the set of acceptable values for the following:

- **issuer;**
- **issuerAltName** (NOTE: If a CIMC does not issue CRLs with this extension, then it is not required within the certificate revocation list profile.)
- **nextUpdate** (i.e., a promise of next CRL in specified time); and
- the set of acceptable CRL and CRL entry extensions.

FMT_MOF_CIMC.6 – RHCS 8.1 provides basic OCSP responses. The Administrator must specify the set of acceptable values (in an OCSP profile) for the following:

- **ResponderID**
- **responseType**

FDP_CIMC_CER.1 – RHCS 8.1 only generates X.509 certificates that meet the following guidelines:

- The **version** field shall contain the integer **0**, **1**, or **2.**
- If the certificate contains an **issuerUniqueID** or **subjectUniqueID** then the **version** field shall contain the integer **1** or **2.**
- If the certificate contains **extensions** then the **version** field shall contain the integer **2.**
- The **serialNumber** shall be unique with respect to the issuing Certification Authority.
- The **validity** field shall specify a **notBefore** value that does not precede the current time and a **notAfter** value that does not precede the value specified in **notBefore.**
- If the **issuer** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical **issuerAltName** extension.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

- If the **subject** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical **subjectAltName** extension.
- The **signature** field and the **algorithm** in the **subjectPublicKeyInfo** field shall contain the OID for a FIPS-approved or recommended algorithm.

Furthermore, RHCS 8.1 performs a POP check before issuing a certificate to ensure that the recipient has the corresponding private key.

FDP_CIMC_CRL.1 – RHCS 8.1 ensures that issued CRLs contain appropriate values. The following items are checked for validity:

- If the **version** field is present, then it shall contain a **1**.
- If the CRL contains any critical extensions, then the **version** field shall be present and contain the integer **1**.
- If the **issuer** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the CRL shall contain a critical **issuerAltName** extension.
- The **signature** and **signatureAlgorithm** fields shall contain the OID for a FIPS-approved digital signature algorithm.
- The **thisUpdate** field shall indicate the issue date of the CRL.
- The time specified in the **nextUpdate** field (if populated) shall not precede the time specified in the **thisUpdate** field.

FDP_CIMC_OCSP.1 – RHCS 8.1 ensures that issued OCSPs contain appropriate values. The following items are checked for validity:

- The **version** field shall contain a **0**.
- If the **issuer** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the response shall contain a critical **issuerAltName** extension.
- The **signatureAlgorithm** field shall contain the OID for a FIPS-approved digital signature algorithm.
- The **thisUpdate** field shall indicate the time at which the status being indicated is known to be correct.
- The **producedAt** field shall indicate the time at which the OCSP responder signed the response.
- The time specified in the **nextUpdate** field (if populated) shall not precede the time specified in the **thisUpdate** field.

6.1.8 Strength of Functions

FCS_SOF_CIMC.1 – RHCS 8.1 is designed to make appropriate use of a FIPS 140-2 certified Hardware Security Module (HSM) for critical cryptographic operations. As such, each of the explicit Strength of Function requirements in section 10 is addressed. The applicable FIPS certificate is as follows: #815.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

7. Protection Profile Claims

As documented in this Security Target (ST), Red Hat Certificate System 8. (RHCS 8.1) is a certificate management system that complies with Certificate Issuing and Management Components (CIMC) In Basic Robustness Environments Protection Profile (PP), Version 1.0, 27 April, 2009 (CIMC-BR-PP).

Conformance to the CIMC-BR-PP is demonstrated as follows:

- The Security Problem Definition in this ST has been reproduced verbatim from the CIMC-BR-PP.
- The Security Objectives in this ST have been reproduced, the the following exceptions, verbatim from the CIMC-BR-PP.
    - FDP_ITT.1 (iteration 3): the qualifier indicating that 'security-relevant' user data was removed so that the SFR in this ST applies to all user data.
    - FDP_ITT.1 (iteration 4): the qualifier indicating that 'confidential' user data was removed so that the SFR in this ST applies to all user data.
    - FPT_ITC.1 (iteration 2): the qualifier indicating that 'confidential' TSF data was removed so that the SFR in this ST applies to all TSF data.
    - FPT_ITT.1 (iteration 3): the qualifier indicating that 'security-relevant' TSF data was removed so that the SFR in this ST applies to all TSF data.
    - FPT_ITT.1 (iteration 4): the qualifier indicating that 'confidential' TSF data was removed so that the SFR in this ST applies to all TSF data.
- The Security Functional Requirements in this ST have been reproduced verbatim from the CIMC-BR-PP.
- The CIMC-BR-PP includes the EAL2 augmented with the ALC_FLR.2 security assurance components. However, this ST conforms with EAL4 augmented with ALC_FLR.2 which is a superset of the assurance required by the CIMC-BR-PP.

Note that the corresponding rationale elements have been referenced in the CIMC-BR-PP since the applicable Security Problem Definition, Objective, and Security Functional Requirement statements have been reproduced in this ST. While the assurance claim has been increased in this ST, that does not affect the rationale as presented in the CIMC-BR-PP.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

8. Rationale

This section includes the rationale for the functional and assurance requirements specified for the TOE. The rationale is based on specified objectives, threats, assumptions, and policies.

8.1 Security Objectives Rationale

The Security Problem Definition and Security Objectives have been drawn directly from the CIMC In Basic Robustness Environments Protection Profile and as such the corresponding rationale is not repeated here.

8.2 Security Requirements Rationale

The Security Objectives and Security Functional Requirements have been drawn directly from the CIMC In Basic Robustness Environments Protection Profile and as such the corresponding rationale is not repeated here.

8.3 Requirement Dependency Rationale

The following table provides a summary of the TOE security functional requirements dependency analysis.

Note that security functional requirements assigned to the IT environment by the CIMC PP are identified in bold-italics. Essentially those dependencies are fulfilled via the security objectives for the TOE environment that correspond to those requirements (see section 4).

Table 8-1 Summary of Security Functional Requirements Dependencies

TOE Component	Dependencies	Fulfilled by:
FAU_GEN.1 (iteration 2)	FPT_STM.1	FPT_STM.1 (iteration 2)
FAU_GEN.2 (iteration 2)	FAU_GEN.1	FAU_GEN.1 (iteration 2)
	FIA_UID.1	FIA_UID.1 (iteration 2)
FAU_SEL.1 (iteration 2)	FAU_GEN.1	FAU_GEN.1 (iteration 2)
	FMT_MTD.1	***FMT_MTD.1***
FAU_STG.1 (iteration 2)	FAU_GEN.1	FAU_GEN.1 (iteration 2)
FAU_STG.4 (iteration 2)	FAU_STG.1	FAU_STG.1 (iteration 2)
FCO_NRO_CIMC.3	FIA_UID.1	FIA_UID.1 (iteration 2)
FCO_NRO_CIMC.4	FCO_NRO_CIMC.3	FCO_NRO_CIMC.3
FCS_CKM_CIMC.5	FCS_CKM.4	***FCS_CKM.4***
	FDP_ACF.1	FDP_ACF.1 (iteration 2)
FCS_SOF_CIMC.1	None	
FDP_ACC.1 (iteration 2)	FDP_ACF.1	FDP_ACF.1 (iteration 2)
FDP_ACF.1 (iteration 2)	FDP_ACC.1	FDP_ACC.1 (iteration 2)
	FMT_MSA.3	***FMT_MSA.3***
FDP_ACF_CIMC.2	None	
FDP_ACF_CIMC.3	None	
FDP_CIMC_CER.1	None	
FDP_CIMC_CRL.1	None	
FDP_CIMC_CSE.1	None	
FDP_CIMC_OCSP.1	None	
FDP_ETC_CIMC.5	None	
FDP_ITT.1 (iterations 3 and 4)	FDP_ACC.1, or FDP_IFC.1	FDP_ACC.1 (iteration 2)
FDP_SDI_CIMC.3	None	
FDP_UCT.1 (iteration 2)	FDP_ACC.1, or FDP_IFC.1	FDP_ACC.1 (iteration 2)

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

TOE Component	Dependencies	Fulfilled by:
	FTP_ITC.1, or FTP_TRP.1	NOT Included (see below)
FIA_SOS.1 (iteration 2)	None	
FIA_UAU.1 (iteration 2)	FIA_UID.1	FIA_UID.1 (iteration 2)
FIA_UID.1 (iteration 2)	None	
FIA_USB.1 (iteration 2)	FIA_ATD.1	***FIA_ATD.1***
FMT_MOF.1 (iteration 2)	FMT_SMR.1	***FMT_SMR.2***
FMT_MOF_CIMC.3	FMT_MOF.1	FMT_MOF.1 (iteration 2)
	FMT_SMR.1	***FMT_SMR.2***
FMT_MOF_CIMC.5	FMT_MOF.1	FMT_MOF.1 (iteration 2)
	FMT_SMR.1	***FMT_SMR.2***
FMT_MOF_CIMC.6	FMT_MOF.1	FMT_MOF.1 (iteration 2)
	FMT_SMR.1	***FMT_SMR.2***
FMT_MTD_CIMC.4	None	
FMT_MTD_CIMC.5	None	
FMT_MTD_CIMC.7	None	
FPT_CIMC_TSP.1	FAU_GEN.1	FAU_GEN.1 (iteration 2)
	FMT_MOF.1	FMT_MOF.1 (iteration 2)
FPT_ITC.1 (iteration 2)	None	
FPT_ITT.1 (iterations 3 and 4)	None	
FPT_STM.1 (iteration 2)	None	

Justification of Unsupported Dependencies Regarding FTP_ITC.1 or FTP_TRP.1

Component FDP_UCT.1 Basic data exchange confidentiality has a direct dependency on FTP_ITC.1 Inter-TSF trusted channel or FTP_TRP.1 Trusted path that is unmet. This product uses basic encryption to ensure basic data exchange confidentiality, as such it is unnecessary for this product to explicitly require Inter-TSF trusted channel or trusted path.

8.4 TOE Summary Specification Rationale

The following table describes the association between the TOE Security Functions and the TOE Security Functional Requirements. This table in conjunction with rationale provided in Section 6.1 demonstrates that the TOE Security Functional Requirements are satisfied.

Table 8-2 Security Function to TOE SFR Mapping

Security Function	Security Functional Components
Identification and authentication	FIA_SOS.1 Verification of secrets
	FIA_UAU.1 Timing of authentication (iteration 2)
	FIA_UID.1 Timing of identification (iteration 2)
	FIA_USB.1 User-subject binding (iteration 2)
Access Control	FDP_ACC.1 Subset access control (iteration 2)
	FDP_ACF.1 Security attribute based access control (iteration 2)
Security Management	FMT_MOF.1 Management of security functions behavior (iteration 2)
Security Audit	FAU_GEN.1 Audit data generation (iteration 2)
	FAU_GEN.2 User identity association (iteration 2)
	FAU_SEL.1 Selective audit (iteration 2)
	FAU_STG.1 Protected audit trail storage (iteration 2)
	FAU_STG.4 Prevention of audit data loss (iteration 2)
	FPT_STM.1 Reliable time stamps (iteration 2)
	FPT_CIMC_TSP.1 Audit log signing event

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Security Function	Security Functional Components
Remote Data Entry & Export	FCO_NRO_CIMC.3 Enforced proof of origin and verification of origin
	FCO_NRO_CIMC.4 Advanced verification of origin
	FDP_CIMC_CSE.1 Certificate status export
	FDP_UCT.1 Basic data exchange confidentiality (iteration 2)
	FDP_ITT.1 Basic internal transfer protection (iterations 3 and 4)
	FPT_ITC.1 Inter-TSF confidentiality during transmission (iteration 2)
	FPT_ITT.1 Basic internal TSF data transfer protection (iterations 3 and 4)
Key Management	FCS_CKM_CIMC.5 CIMC private and secret key zeroization
	FDP_ACF_CIMC.2 User private key confidentiality protection
	FDP_ACF_CIMC.3 User secret key confidentiality protection
	FDP_ETC_CIMC.5 Extended user private and secret key export
	FDP_SDI_CIMC.3 Stored public key integrity monitoring and action
	FMT_MTD_CIMC.4 TSF private key confidentiality protection
	FMT_MTD_CIMC.5 TSF secret key confidentiality protection
	FMT_MTD_CIMC.7 Extended TSF private and secret key export
Certificate Management	FDP_CIMC_CER.1 Certificate Generation
	FDP_CIMC_CRL.1 Certificate Revocation
	FDP_CIMC_OCSP.1 Basic Response Validation
	FMT_MOF_CIMC.3 Extended certificate profile management
	FMT_MOF_CIMC.5 Extended certificate revocation list profile management
	FMT_MOF_CIMC.6 OCSP Profile Management
Strength of Functions	FCS_SOF_CIMC.1 CIMC Strength of Functions

8.5 PP Claims Rationale

As indicated in Section 7, Red Hat Certificate System 8.1 (RHCS 8.1) complies with Certificate Issuing and Management Components (CIMC) In Basic Robustness Environments Protection Profile (PP), Version 1.0, April 27, 2009.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

9. Access control policies

9.1 CIMC TOE Access Control Policy

The TOE shall support the administration and enforcement of a CIMC TOE access control policy that provides the capabilities described below.

Subjects (human users) will be granted access to objects (data/files) based upon the:

1. Identity of the subject requesting access,
2. Role (or roles) the subject is authorized to assume,
3. Type of access requested,
4. Content of the access request, and,
5. Possession of a secret or private key, if required.

Subject identification includes:

- Individuals with different access authorizations
- Roles with different access authorizations
- Individuals assigned to one or more roles with different access authorizations

Access type, with explicit allow or deny:

- Read
- Write
- Execute

For each object, an explicit owning subject and role will be identified. Also, the assignment and management of authorizations will be the responsibility of the owner of an object or a role(s), as specified in this PP.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

10. Strength of Function (SoF) Requirements

This section defines explicit metrics for various cryptographic functions in support of the FCS_SOF_CIMC.1 SFR.

10.1 Cryptographic Modules

FIPS 140-2 validated cryptographic modules must perform all cryptographic functions performed by CIMCs. FIPS 140-2 validated cryptographic modules are also required to generate cryptographic keys and to store plaintext private and secret keys.

10.1.1 Encryption and FIPS 140-2 Validated Modules

As noted earlier in the document, references to FIPS 140-2 refer to the most current version of the standard and the most current version can be found at http://csrc.nist.gov/cryptval.

10.1.1.1 Encryption Algorithms

The encryption specified for:

Requirement Label	Requirement Name	Encryption Summary
FAU_STG.1	Protected audit trail storage	Not applicable – access controlled
FCO_NRO_CIMC.4	Advanced verification of origin	CRL signing: default CA cert 2048-bit RSA; OCSP signing: default 2048-bit RSA; SSL server: default 2048-bit RSA. All configurable to RSA (1024-, 2048-, 3072-, 4096-, 8192-bits and others supported by the HSM).
FDP_ACF_CIMC.2	User private key confidentiality protection	DRM storage cert: default to 2048-bit RSA, configurable to various key sizes (1024-, 2048-, 3072-, 4096-, 8192-bits and others supported by the HSM). Symmetric session key: 3DES
FDP_ACF_CIMC.3	User secret key confidentiality protection	Not applicable – no user secret keys stored
FDP_ETC_CIMC.5	Extended user private and secret key export	DRM transport cert: default to 2048-bit RSA, configurable to various key sizes (1024-, 2048-, 3072-, 4096-, 8192-bits and others supported by the HSM). Symmetric session key: 3DES
FDP_SDI_CIMC.3	Stored public key integrity monitoring and action	Certificate signing: default CA cert 2048-bit RSA, configurable to RSA (1024-, 2048-, 3072-, 4096-, 8192-bits and others supported by the HSM).
FMT_MTD_CIMC.4	TSF private key confidentiality protection	TSF private keys are stored and

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

		protected on the HSM.
FMT_MTD_CIMC.5	TSF secret key confidentiality protection	TSF secret keys are stored and protected on the HSM.
FMT_MTD_CIMC.7	Extended TSF private and secret key export	Not applicable – no TSF private or secret keys exported
FPT_CIMC_TSP.1	Audit log signing event	Default signing cert key: 2048-bit RSA, but configurable to other sizes (1024-, 2048-, 3072-, 4096-, 8192-bits and others supported by the HSM).

shall be performed using a FIPS-approved or recommended algorithm.

10.1.1.2 FIPS 140-2 Validated Cryptographic Modules

Cryptographic modules specified for:

FDP_ACF_CIMC.2	User private key confidentiality protection
FDP_ACF_CIMC.3	User secret key confidentiality protection
FDP_ETC_CIMC.5	Extended user private and secret key export
FDP_SDI_CIMC.3	Stored public key integrity monitoring and action
FMT_MTD_CIMC.4	TSF private key confidentiality protection
FMT_MTD_CIMC.5	TSF secret key confidentiality protection
FMT_MTD_CIMC.7	Extended TSF private and secret key export
FPT_CIMC_TSP.1	Audit log signing event

shall be validated against FIPS 140-2.

10.1.1.3 Split Knowledge Procedures

Split-knowledge procedures specified in:

FDP_ETC_CIMC.5	Extended user private and secret key export
FMT_MTD_CIMC.7	Extended TSF private and secret key export

shall be implemented and validated as specified in FIPS 140-2.

10.1.1.4 Authentication Codes

The authentication code specified in:

FAU_STG.1	Protected audit trail storage
FCO_NRO_CIMC.4	Advanced verification of origin
FPT_CIMC_TSP.1	Audit log signing event
FDP_SDI_CIMC.3	Stored public key integrity monitoring and action

shall be a FIPS-approved or recommended authentication code.

10.1.2 Cryptographic module levels for cryptographic functions that involve private or secret keys

All cryptographic operations performed (including key generation) at the request of the TOE shall be performed in a FIPS 140-2 validated cryptographic module operating in a FIPS-approved or recommended mode of operation.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Table 10-1specifies for each category of use for a private or secret key, the required overall FIPS 140-2 level for the validated cryptographic module. If the CIMC generates certificate subject private keys, the required overall FIPS 140-2 level for *Long Term Private Key Protection* keys shall apply.

Table 10-1 FIPS 140-2 Level for Validated Cryptographic Module

Required Overall FIPS 140-2 Level for CIMC Cryptographic Modules	
Category of Use	FIPS 140-2 Level
Certificate and Status Signing - single party signature - multiparty signature	 3 2
Integrity or Approval Authentication - single approval - dual approval	 2 2
General Authentication	2
Long Term Private Key Protection	3
Long Term Confidentiality	2
Short Term Private key Protection	2
Short Term Confidentiality	1

10.1.3 Cryptographic Functions That Do Not Involve Private or Secret Keys

There are two other cryptographic functions that may be performed in CIMCs that do not require private or secret keys. These include:

1. *Hash Generation*: One-way hash functions may be used in the process of signature generation and verification (a signature is typically generated by applying a private key to the hash of the message). The generation of a hash does not require a key. Therefore, hash generation does not have the same confidentiality requirements of other cryptographic functions.

2. *Signature Verification*: Signatures are verified from a message text and a public key.

For a cryptographic module that only performs signature verification and/or keyless hash generation functions, the overall required FIPS 140-2 level shall be Level 1.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

11. Glossary of terms

The following definitions are used throughout this standard:

Authentication code: a cryptographic checksum, based on a FIPS-approved or recommended security method; also known as a Message Authentication Code (MAC) in ANSI standards.

CIMC: the set of hardware, software, firmware, or some combination thereof, that issues, revokes, and manages public key certificates and certificate status information, and is contained within the CIMC boundary.

CIMC boundary: an explicitly defined contiguous perimeter that establishes the physical bounds of a CIMC.

Compromise: the unauthorized disclosure, modification, substitution or use of sensitive data (including plaintext cryptographic keys and other CSPs).

Confidentiality: the property that sensitive information is not disclosed to unauthorized individuals, entities or processes.

Critical security parameter (CSP): security-related information (e.g., secret and private cryptographic keys, authentication data such as passwords and PINs) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a CIMC or the security of the information protected by the CIMC.

Cryptographic key (key): a parameter used in conjunction with a cryptographic algorithm that determines:

- the transformation of plaintext data into ciphertext data,
- the transformation of ciphertext data into plaintext data,
- a digital signature computed from data,
- a keyed hash computed from data,
- the verification of a digital signature computed from data,
- an authentication code computed from data, or
- an exchange agreement of a shared secret.

Cryptographic key component (key component): a parameter used in conjunction with other key components in a FIPS-approved or recommended security method to form a plaintext cryptographic key or perform a cryptographic function.

Digital signature: a non-forgeable transformation of data that allows proof of the source (with non-repudiation) and verification of the integrity of that data.

Encrypted key: a cryptographic key that has been encrypted with a key encrypting key, a PIN or a password in order to disguise the value of the underlying plaintext key.

Error detection code (EDC): a code computed from data and comprised of redundant bits of information designed to detect, but not correct, unintentional changes in the data.

FIPS-Approved or recommended mode of operation: a mode that employs only the operation of FIPS-approved or recommended security methods.

FIPS-approved or recommended security method: a security method (e.g., cryptographic algorithm, cryptographic key generation algorithm or key distribution technique, authentication technique, or evaluation criteria) that is either a) specified in a FIPS or b) adopted in a FIPS and specified either in an appendix to the FIPS or in a document referenced by the FIPS.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Firmware: the programs and data stored in hardware (e.g., ROM, PROM, or EPROM) such that the programs and data cannot be dynamically written or modified during execution. *Hardware*: the physical equipment used to process programs and data in a CIMC.

Integrity: the property that sensitive data has not been modified or deleted in an unauthorized and undetected manner.

Key encrypting key: a cryptographic key that is used for the encryption or decryption of other keys.

Key management: the activities involving the handling of cryptographic keys and other related security parameters (e.g., IVs, passwords) during the entire life cycle of the keys, including their generation, storage, distribution, entry and use, deletion or destruction, and archiving.

Password: a string of characters (letters, numbers, and other symbols) used to authenticate an identity or to verify access authorization.

Personal Identification Number (PIN): a 4 or more character alphanumeric code or password used to authenticate an identity, commonly used in banking applications.

Physical protection: the safeguarding of a CIMC, cryptographic keys, or other CSPs using physical means.

Plaintext key: an unencrypted cryptographic key.

Private key: a cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and not made public.

Protection Profile: an implementation-independent set of security requirements for a category of Targets of Evaluation (TOEs) that meet specific consumer needs.

Public key: a cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and which may be made public. (Public keys are not considered CSPs.)

Public key certificate: a set of data that unambiguously identifies an entity, contains the entity's public key, is digitally signed by a trusted party, and binds the public key to the entity.

Public key (asymmetric) cryptographic algorithm: a cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that, given the public key, it is computationally infeasible to derive the private key.

Secret key: a cryptographic key used with a secret key cryptographic algorithm, uniquely associated with one or more entities, and which shall not be made public. The use of the term "secret" in this context does not imply a classification level rather the term implies the need to protect the key from disclosure or substitution.

Secret key (symmetric) cryptographic algorithm: a cryptographic algorithm that uses a single, secret key for both encryption and decryption.

Security policy: a precise specification of the security rules under which a CIMC shall operate, including the rules derived from the requirements of this document and additional rules imposed by the vendor.

Software: the programs and associated data that can be dynamically written and modified.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

Split knowledge: a condition under which two or more entities separately have key components that individually convey no knowledge of the plaintext key that will be produced when the key components are combined in the cryptographic module.

Target of Evaluation (TOE) - An information technology product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.

TOE Security Functions (TSF) - A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

TOE Security Policy (TSP) - A set of rules that regulate how assets are managed, protected and distributed within a TOE.

Trusted path: a means by which an operator and a TSF can communicate with the necessary confidence to support the TSP.

User: an individual, or a process (subject) operating on behalf of the individual, accessing CIMC.

Zeroization: a method of erasing electronically stored data by altering or deleting the contents of the data storage so as to prevent the recovery of the data.

© 2009-2012 Red Hat, Inc.
All Rights Reserved.

12. Acronyms

ANSI	American National Standards Institute
CA	Certification Authority
CC	Evaluation Criteria for Information Technology Security (Common Criteria)
CIMC	Certificate Issuing and Management Component
CIMS	Certificate Issuing and Management System
CS	Certificate System
CP	Certificate Policy
CPS	Certification Practices Statement
CRL	Certificate Revocation List
DRM	Data Recovery Manager (formerly KRA)
EAL	Evaluation Assurance Level
I&A	identification and authentication
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
IT	Information Technology
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
JSS	Java Security Services
NSS	Network Security Services
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PKI	Public Key Infrastructure
POP	Proof of Possession
PP	Protection Profile
RA	Registration Authority
RHCS 8.1	Red Hat Certificate System 8.1
SFP	Security Function Policy
SSL	Secure Socket Layer
ST	Security Target
TKS	Token Key Service
TMS	Token Management System
TOE	Target of Evaluation
TPS	Token Processing System
TSF	TOE Security Functions
TSP	TOE Security Policy

© 2009-2012 Red Hat, Inc.
All Rights Reserved.