RedCastle v3.0 for Asianux Server 3 Security Target

Document ID	RCX3-07D-ASE-04
Document Version	Version 1.4
Author	RedGate Co., Ltd.
Date of generation	30 Jan, 2008

Document History

Document No.	Summary	Date	Author
RCX3-07D-ASE-01	Initial version of ST - RedCastle v3.0 for Asianux	1 Aug 2007	S.C. Kim
RCX3-07D-ASE-02	Updates to the TOE description	10 Sep 2007	S.C. Kim
RCX3-07D-ASE-03	1st OR updates	17 Dec 2007	S.C. Kim
RCX3-07D-ASE-04	2nd OR updates	30 Jan 2008	S.C. Kim

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1 ST identification

This is the Security Target (ST) of the RadCastle v3.0 for Asianux Server 3, which is an Secure Operating System operating in the form of software on Asianux Server 3 OS.

- Title: RedCastle v3.0 for Asianux Server 3 Security Target
- Document No.: RCX3-07D-ASE-04
- Document version: Version 1.4
- Author: RedGate Co., Ltd.
- Date of issue: 30 Jan 2008
- Product name: RedCastle v3.0 for Asianux
- TOE name: RedCastle v3.0 for Asianux Server 3
- PP claim: N/A
- Referenced standard: Common criteria for information technology security evaluation (Notification no.2005-25 of the MIC), CC v2.3
- Assurance level: EAL4
- Keyword: Secure Operating System, LBAC, RBAC

1.2 ST overview

This ST describes the security features of the RedCastle v3.0 for Asianux.

RedCastle v3.0 for Asianux provides LBAC, RBAC, and functions to control the user login service and login session to reinforce the security of Asianux Server 3 (AXS3 hereinafter).

RedCastle v3.0 for Asianux is comprised of RedCastle Agent and RedCastle Manager. RedCastle Agent is installed and operating on Asianux Server 3, providing main functions such as:

- LBAC, RBAC, allow/deny list-based DAC

- User login service control, login session control
- Security management function to manage control policies

RedCastle Manager is a security management program that provides a GUI for united management of multiple RedCastle Agents. It is installed in Windows XP Professional SP2 and shall be connected through network to Asianux Server 3 system in which RedCastle Agent is installed.

This ST specifies the IT security functions and assurance measures of the TOE, while the security of Asianux Server 3 is not covered.

1.3 CC conformance

This ST conforms to the following standards:

- CC (notification no.2005-25 by the MIC) v2.3, Part 2
- CC v2.3, Part 3
- EAL4 from the CC v2.3, Part 3

The strength of function (SOF) claimed in this TOE is SOF-basic.

1.4 ST contents

Chapter 1 – ST Introduction – identifies the ST and gives general information.

Chapter 2 – TOE Description – defines the TOE and describes its physical and logical scope.

Chapter 3 – TOE Security Environment – describes the security problem of the TOE and its environment in terms of assumptions, threats, and organizational security policies.

Chapter 4 – Security Objectives – describes the security objectives for the TOE and its environment that counter the threats identified in TOE Security Environment and cover the assumptions and OSPs.

Chapter 5 – IT Security Requirements – describes security functional requirements and assurance requirements that are necessary to satisfy the security objectives.

Chapter 6 – TOE Summary Specification – describes the security functional requirements and assurance requirements that satisfy the IT security requirements.

Chapter 7 – PP Claims – describes the protection profile to which this ST claims conformance.

Chapter 8 – Rationale – demonstrates that the TOE provides effective IT security measures in its security environment; gives rationale for the security objectives, security requirements, TOE summary specification, PP claims, and SOF claim.

1.5 Conventions

This ST uses both Korean and English for some abbreviations or to communicate a clear meaning. Notations, forms, conventions used conform to the Common Criteria for information technology security evaluation (notification no.2005-25 by the MIC, hereinafter CC).

Permitted operations on the security functional requirements in the CC are selection, assignment, refinement, and iteration. Refer to the CC v2.3 Part 2, 6.4.1.3.2.

Iteration

The use of a component more than once with varying operations. The result of iteration operation is marked by adding a figure in a parenthesis, (the number of iteration), next to the component identifier.

Selection

The specification of one or more items from a list in a component. The result is marked by *underlined italics*.

Refinement

The addition of details to a requirement, thus further limiting it. The result is marked by **bold letters**.

Assignment

The specification of a parameter, for example a password length. The result is marked with the value in a square bracket, e.g., [assignment_value].

Application notes will be given with the requirement if necessary to clarify the meaning of requirements, provide information about the options in implementation, and to define the standard of deciding “conformance/non-conformance” of requirements.

1.6 Terminology

The same terms used in this ST and the CC at the same time have the same meanings as in the CC.

Mandatory Access Control (MAC)

A means to control access to objects based on the sensitivity label of an object and the authority of a subject to access.

Label-based Access Control (LBAC)

A kind of MAC where the security attributes of a subject and object have a label by the security level and category; the label gives ground to mandatorily control the access of that subject to that object. BLP model is an example. An abbreviated form of label-based mandatory access control.

Bell-LaPadula Model (BLP model)

A MAC model devised by David Elliott Bell and Len LaPadula, which secures confidentiality of information based on its label.

Object

An entity within the TSC that contains or receives information and upon which subjects perform operations.

Attack potential

The perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker's expertise, resources and motivation.

Strength-of-Function (SOF)

A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behavior by directly attacking its underlying security mechanisms.

Security Target (ST)

A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.

Security label

A combination of a security level (hierarchical classification) and security category (non-hierarchical category), which labels the sensitivity of a user or information.

Protection Profile (PP)

An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.

User

A human who interacts with the TOE.

Identity

A representation (e.g. a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Element

An indivisible security requirements.

Role

Authorities and responsibilities assigned to a personnel or user; which is a basic means to establish the relationship between a user and permission to access.

Role Based Access Control (RBAC)

A means to control the access of a user to an object with the roles that are dependent on the property of an organization as a mediator, thus based on the 'user-role relationship' and 'access permission-role relationship' rather than a direct relationship between the user and access permission. It may be applied by using Core RBAC, hierarchical model, and separation of duties model.

Operation

A set of calculation or action defined by a computer instruction or pseudo-instruction.

Threat agent

An unauthorized user or external IT entity that causes threats for assets such as illicit access, modification, or deletion.

External IT entity

Any IT product or system, untrusted or trusted, outside of the TOE that interacts with the TOE.

Authorized administrator

An authorized user who is granted the authority to manage the TOE.

Authorized user

A user who may, in accordance with the TSP, perform an operation.

Authentication data

Information used to verify the claimed identity of a user.

Discretionary Access Control (DAC)

A means to control access to objects based on the identity of a user or group to which the user belongs.

Assets

Information or resources to be protected by the countermeasures of a TOE.

Common criteria for information technology security evaluation (CC)

The common criteria (CC) is meant to be used as the basis for evaluation of security properties of IT products and systems. It comprises existing criteria from different countries to develop criteria that can be accepted and applied everywhere with a common language and understanding. The CC v2.3 was translated into Korean and announced by the Minister of Information and Communication on 21 May 2005.

Organizational Security Policies (OSP)

One or more security rules, procedures, practices, or guidelines imposed by an organization upon its operations.

Dependency

A relationship between requirements such that the requirement that is depended upon must normally be satisfied for the other requirements to be able to meet their objectives.

Subject

An entity within the TSC that causes operations to be performed.

Sensitivity Label

Security attributes that indicates the security label of a subject or object.

Abstract Machine

A known or evaluated combination of hardware/software, which executes as the platform of hardware/firmware or a virtual machine. If a TOE is application program, the underlying abstract machine will be the Operating System; and if a TOE is operating system, it will be the firmware or hardware.

Component

The smallest selectable set of elements that may be included in a PP, an ST, or a package.

Class

A grouping of families that share a common focus.

Target of Evaluation (TOE)

An IT product or system and its associated guidance documentation that is the subject of an evaluation.

Evaluation assurance level (EAL)

A package consisting of assurance components from Part 3 that represents a point on the CC predefined assurance scale.

Core RBAC

An RBAC model that defines its basic concepts, including the requirements on many-to-many relationship of the user-role and access permission-role assignment.

TOE Security Function (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.

TSF Data

Data created by and for the TOE, that might affect the operation of the TOE.

TSF Scope of Control (TSC)

The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.

Console

Also called “system console,” a keyboard and monitor connected to a computer. “Console login” refers to a way to access a computer through a console, which normally is more trusted by a computer OS than the remote login through network of the computer.

RedCastle

The representative product name produced by RedGate.

RedCastle Manager

The security management part of RedCastle to manage the security functions of RedCastle Agent. Operates in the Windows XP Professional SP2 system and provides GUI for an administrator.

RedCastle Agent

The security function processing part of RedCastle. Installed and operated on Asianux Server 3 and performs a variety of security functions including access control.

RedCastle Java Manager (or Java Manager)

The security management part of RedCastle implemented in Java. Installed and operated on the server on which RedCastle Agent is installed and provides GUI for the management of security functions.

2 TOE description

2.1 Product type

RedCastle v3.0 for Asianux is Secure Operating System devised for Asianux Server 3. It provides

- Label-based Access Control (LBAC) that assigns label to the user and resources and upon which applies MAC policy;

- Role-based Access Control (RBAC) that defines the role of a user and assigns subject information to the role to apply RBAC policy; and

- DAC based on allow/deny list such as command control list, SETUID file execution permission list, KILL prevention process list, permitted su user list, etc.

to enhance the security of OS.

RedCastle v3.0 for Asianux also provides PAM (Pluggable Authentication Module) account module for the service control function that restricts a user login based on the user's identity, service name, login time, and access IP and PAM session module that restricts the number of login sessions.

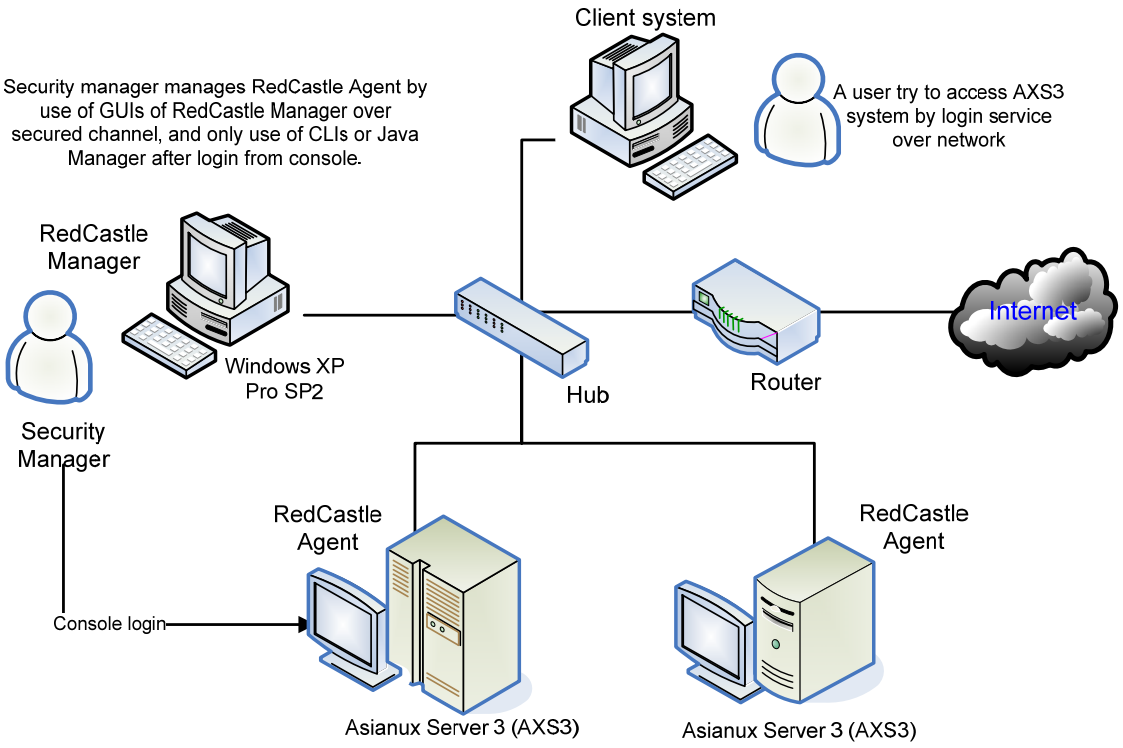
AXS3 OS is a kind of Linux distribution that supports multi-user and multi-tasking officially announced in September 2007 by Asianux Consortium, which has been developing and selling Asianux OS since founded in December 2003 by Haansoft (Korea), Red Flag Software (China), and Miracle Linux (Japan).

RedCastle v3.0 for Asianux is an Secure Operating system officially proposed by Asianux Consortium and adopted for the security enhancement of AXS3. AXS3 is developed based on the Linux Kernel v2.6.18, of which the official Kernel version is 2.6.18-8.10AX.

The target of evaluation in this ST is RedCastle v3.0 for Asianux Server 3, which comprises RedCastle Agent (the security function processing part) and RedCastle Manager (the security management part). RedCastle Agent is installed and operated on Asianux Server 3 and RedCastle Manager on Windows XP Professional SP2.

These two systems on which RedCastle is installed should be connected through network. RedCastle Manager can be managed by a number of RedCastle Agents. Between RedCastle Manager and RedCastle Agent is established a secure communication channel by SSL protocol.

The structure and configuration of the TOE can be shown as:



[Figure 2-1] RedCastle system configuration

The administrator (i.e., Security Manager) can manage a number of RedCastle Agents by using GUI provided by RedCastle Manager. Besides system user I&A, administrator role identification and separate security password authentication are required when the administrator accesses RedCastle Agent through RedCastle Manager.

The administrator can also perform the security management function with command interface provided by RedCastle Agent and RedCastle Java Manager (a security management tool that is implemented in Java and operates in X Windows

of AXS3) after logged in through the console of Asianux Server 3. The command interface and Java Manager are permitted only for the administrator who logged in through console to operate.

A user logs in through telnet or the like from a client system connected to network to use the application service of AXS3 on which RedCastle Agent is installed, when the service control and session control functions decide whether to allow the user to use the service. When the allowed user accesses the object (file or process) of AXS3 system, the access control function of RedCastle Agent applies to decide whether to permit the access. The service control and session control policies equally apply when the administrator logs in through console.

The user service and session control does not apply when the user intends to access an object through service provided by network (e.g. Web service, database service) rather than tries direct access after login. It applies however when Web service being executed on behalf of the user accesses the object.

2.2 Physical scope and boundaries of the TOE

The TOE is RedCastle v3.0 for Asianux Server 3, software comprising RedCastle Agent (the security function processing part) and RedCastle Manager (the security management part).

RedCastle Agent is comprised of the application that performs security functions such as security management, security audit, and service control and the kernel that performs LBAC, RBAC, and list-based DAC. RedCastle Manager provides GUI so that the Security Manager can perform security management functions such as the TOE security function control, security policy management, and security audit reference.

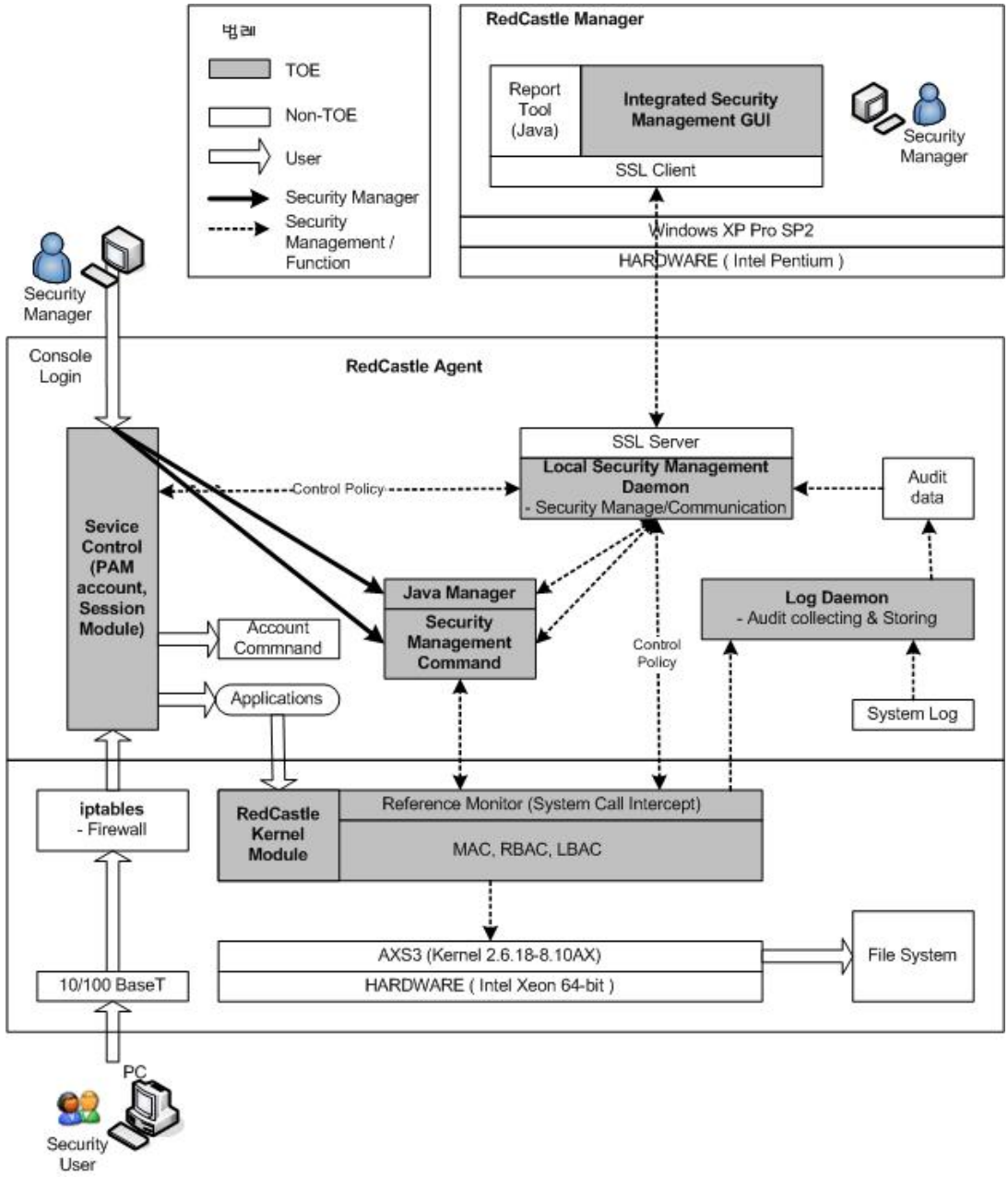
Hardware and software specification required for the operation of the TOE is:

[Table 2-1] Operational environment of the TOE

Item	RedCastle Agent	RedCastle Manager
Software	Asianux Server 3 (Kernel 2.6.18-8.10AX)	Windows XP Professional SP2

Hardware	CPU: Intel Xeon 64 bit 2.4 GHz and above RAM: 1024 MB and above HDD: 500 MB and above Network: 10/100 BaseT	CPU: Pentium IV 1.0 GHz and above RAM: 512 MB and above HDD: 100 MB and above Network: 10/100 BaseT

The physical scope and boundaries of the TOE is shown below:



[Figure 2-2] Physical TOE scope

The Security Manager can perform security management functions such as the TOE security function control, security policy management, and security audit reference through the GUI. The Security Manager can access the system console of Asianux Server 3 on which RedCastle Agent is installed and control and manage the security functions using security management commands. The Security Manager can also use functions such as security module control, security module operational mode and configuration, and RBAC policy reference by using Java Manager in X Windows. Using security management command or Java Manager from remote access that is not by console login is not allowed.

A system user can access the 10/100 BaseT network interface through TCP/IP after passing the filtering by iptables that Asianux Server 3 provides and ID/password authentication. When the user is authenticated successfully, the service control function for user access IP, service, time, and session is provided through PAM.

The user identified and authenticated by the service control is given the security attributes that the RedCastle Kernel module assigns to a user. The module provides reference monitor function and access control function. When the user accesses the file system through applications such as telnet, ftp daemon, etc., or performs other applications, the system call is intercepted by the reference monitor if it is controlled by the TOE, and access to the file is restricted by the access control function.

The access control module performs MAC, RBAC, and DAC functions in accordance with the security policies established by the Security Manager.

The local security management daemon in RedCastle Agent actually performs the security management functions that RedCastle Manager and Java Manager require. Those management functions include management functions in the TOE scope such as access control policy management, service control policy management, and log daemon policy management and also management functions outside the TOE scope such as iptables policy management and system account management.

Log Daemon collects and manages the TOE audit data generated by local security management daemon, access control module, service control module, and log daemon, and iptables log and system log that are generated outside the TOE. The time of occurrence of an event means a reliable timestamp that the TSF intends to use. The TOE uses the timestamp Asianux Server 3 provides.

2.3 Logical scope and boundaries of the TOE

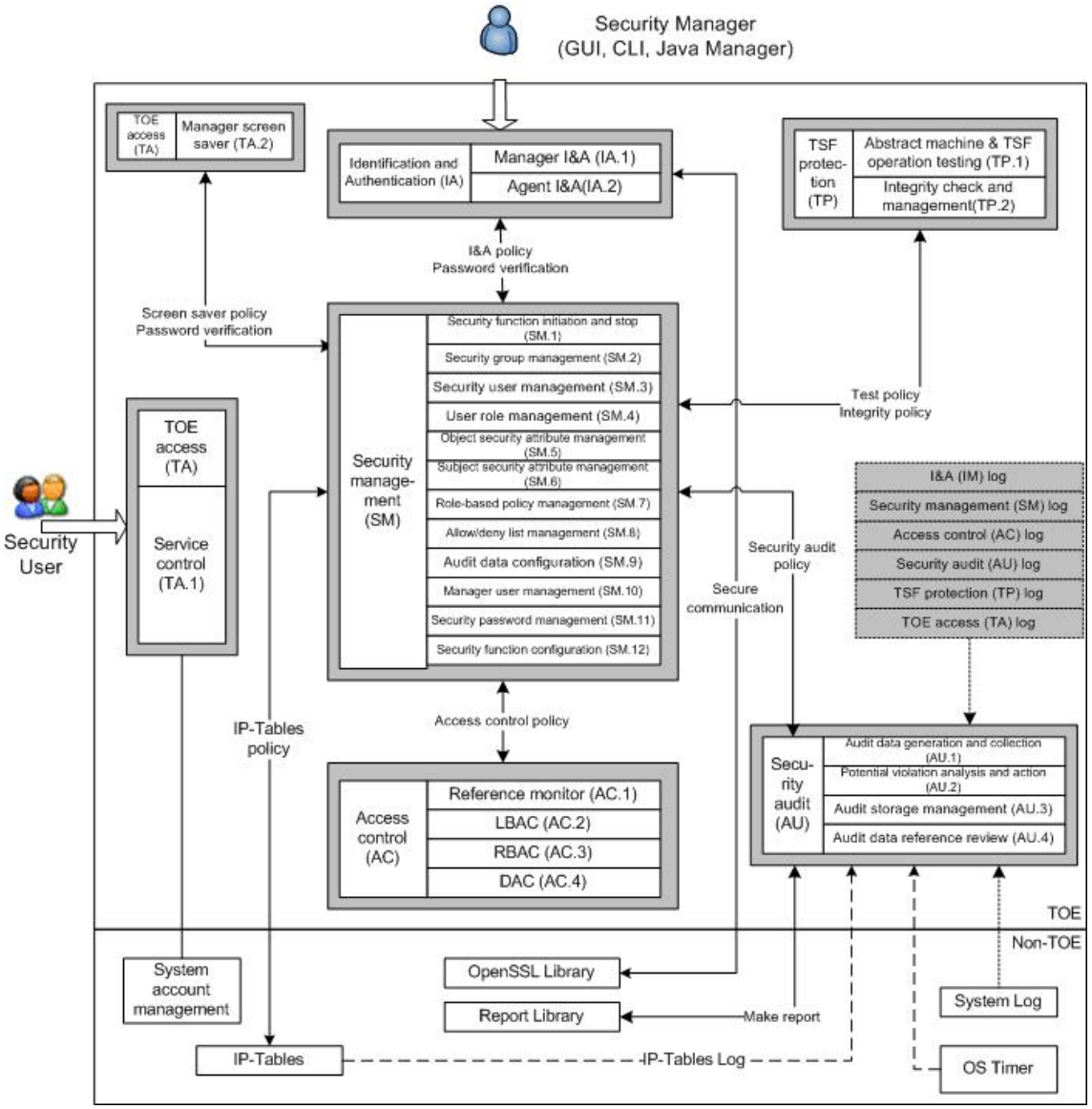
The logical scope of RedCastle v3.0 for Asianux Server 3 includes the following functions:

- Security audit (AU)
- Identification and authentication (IA)
- Access control (AC)
- Security management (SM)
- TSF protection (TP)
- TOE access (TA)

The functions provided by the IT environment are not considered included in the TOE scope. Below are the components provided as the IT environment for the TOE security functions.

- Report Library used by the report function of RedCastle Manager
- SSL Library that provides secure communication between RedCastle Manager and RedCastle Agent
- Iptables module provided by the OS to control the network
- OS account-relevant commands to manage the account
- OS timer to provide time information for audit data generation

[Figure 2-3] is the logical scope and boundaries of the TOE.



[Figure 2-3] Logical TOE scope

2.3.1 Security Audit (AU)

Security audit (AU) functions include Audit data generation and collection (AU.1), Potential violation analysis and action (AU.2), Audit storage management (AU.3), and Audit data reference and review (AU.4).

Audit data generation and collection (AU.1) collects the TOE audit data generated by Identification and authentication (IA), Security management (SM), Access control (AC), Security audit (AU), TSF protection (TP), and TOE access (TA).

Potential violation analysis and action (AU.2) detects if the number of security violation reviewed among the collected audit data exceeds predefined accumulated number per unit time, and gives the alarm.

Audit storage management (AU.3) provides the file size of 10 MB and number of 5 as the default audit storage and takes action according to the status of storage exhaustion to protect the audit trail. It also protects stored audit records from unauthorized deletion or modification and restricts the right to read the audit record only to the Security Manager.

Audit data reference and review (AU.4) provides the Security Manager with the functions to refer to or review the stored TOE audit data and to search and sort by category through the GUI provided by RedCastle Manager. It also provides a function to make a analysis report on the security violation events in the TOE audit data.

2.3.2 Identification and Authentication (IA)

Identification and authentication (IA) functions include Manager identification and authentication (IA.1) and Agent identification and authentication (IA.2).

Manager identification and authentication (IA.1) is for the Manager administrator and user (“Manager User”) when operating RedCastle Management GUI. It performs identification and authentication using Manager ID and password.

Agent identification and authentication (IA.2) is for the Manager user to access each server on which RedCastle Agent is installed through GUI. It performs identification and authentication using system user ID, system password, and security password.

The access types between the Manager and Agent are the management mode that the user can perform security management like establishment of security policies and all security functions including log reference; and the monitoring mode that only the functions of real-time log reference and log search are provided.

2.3.3 Access Control (AC)

Access control (AC) functions include Reference monitor (AC.1), LBAC (AC.2), RBAC (AC.3), and DAC (AC.4).

Reference monitor (AC.1) performs the process of adding system calls used by the TOE and replacing the system call of the OS when the security functions are being launched and also performs the other way around as the process of recovering and deleting the system call.

LBAC (AC.2) provides the rules by which the security attributes are given and inherited and the MAC dependent on the security attributes of the subject and object. The TOE applies BLP model transformed into LBAC model.

RBAC (AC.3) enforces access control rules based on the subject's role and object's access permission and forms ACL. "User role" refers to the role assigned to the user. A role-based policy allows only one role for a user.

DAC (AC.4) is comprised of the access control based on allow/deny list, and allows the addition of the operations that are not on the ACL or requires an exceptional management. The TOE provides allow/deny lists, which include SETUID execution permission list, permitted su user list, command control list, and kill prevention process list.

2.3.4 Security Management (SM)

Security management (SM) functions include Security function initiation and stop (SM.1), Security group management (SM.2), Security user management (SM.3), User role management (SM.4), Object security attribute management (SM.5), Subject security attribute management (SM.6), Role-based policy management (SM.7), Allow/deny list management (SM.8), Audit data configuration (SM.9), Manager user management (SM.10), Security password management (SM.11), Security function configuration (SM.12).

Security function initiation and stop (SM.1) provides the Security Manager with the function to initiate or stop actions of security function processing in the OS boot or shutdown, or when exceptional duties such as system maintenance are required. It also provides Warning or Enable mode of the security function.

Security group management (SM.2) provides the functions to refer to, add, delete, and modify the security group (i.e. security category), which is non-hierarchical, attribute of the sensitivity label of a subject or object.

Security user management (SM.3) can give a user the sensitivity label and refer to, modify, and delete the security attribute of the security user.

User role management (SM.4) provides the functions to refer to, generate, and delete the user's role in order to manage the role that is necessary for the RBAC policies.

Object security attribute management (SM.5) can give a file, an object, the sensitivity label and refer to, modify, and delete the security attribute of the file.

The TOE is designed to assign the security attribute to a user automatically when a new process is generated, where the Security Manager uses Subject security attribute management (SM.6)

Role-based policy management (SM.7) provides the functions to refer to, add, delete, and modify the role-based policy group and the role-based policies based on the ACL.

Allow/deny list management (SM.8) can add, refer to, or delete the command control list, kill prevention process list, setuid execution permission list, and permitted su user list.

Audit data configuration (SM.9) provides the Security Manager with the functions to set the place, size, and alarm of the audit storage.

Manager user management (SM.10) provides the functions to add and delete the Manager User and to modify the Manager password through the GUI provided by RedCastle Manager.

Security password management (SM.11) provides the functions to register and modify the authentication data (i.e. security password) of the Security Manager.

Security function configuration (SM.12) provides the functions to set On, Warning, and Off of the operational environment of the security function.

2.3.5 TSF Protection (TP)

TSF protection (TP) functions include Abstract machine and TSF operation testing (TP.1) and Integrity check and management (TP.2).

Abstract machine and TSF operation testing (TP.1) provides the functions to test the operation of the abstract machine and TSF in order to examine the operating status of the TOE and the system on which RedCastle Agent is installed.

Integrity check and management (TP.2) provides the functions to check the integrity of the TSF execution file, TSF data file, and file the administrator requires at the start-up, periodically during normal operation, or at the request of the Security Manager to ensure the secure operation of the TSF.

2.3.6 TOE Access (TA)

TOE access (TA) functions include Service control (TA.1) and Manager screen saver (TA.2).

Service control (TA.1) provides the functions of login service control and session control based on the PAM. For the user who accesses Asianux Server 3 through the login service, the login service permission is decided by PAM account module of the TOE and the login session permission is by PAM session module of the TOE. That is, all users that access Asianux Server 3 are controlled by the PAM module of the TOE. The TOE provides the management functions of the service control policy and session control policy by the user and group, default service control policy, and default session policy for the service control.

Manager screen saver (TA.2) provides the functions to lock the interacting session after the Security Manager's inactivity and to unlock the session using the RedCastle Manager's identification and authentication.

2.3.7 What is not the TOE

The functions provided by the IT environment are not included in the TOE scope. Those components provided as the IT environment for the TOE security function and the non-security functions that are outside the TOE scope will be described below.

The TSF needs a reliable timestamp in generating the TOE audit data, where the timer of Asianux Server 3, which is the IT environment, is used. The TSF obtains the time information by calling a standard library function.

- Time-related function call of ANSI C: time()

The TOE uses SSL version 3 protocol for the secure communication between RedCastle Manager and RedCastle Agent. The SSL protocol is provided as the IT environment and uses openssl 0.9.7c without modification. The key exchange and authentication method of the SSL protocol is anonymous authentication mechanism (anonymous DH); AES cryptographic algorithm (256 bit encryption key) and SHA hash algorithm are selected for the cipher-suite (ADH-AES256-SHA). Authentication data is protected as the SSL protocol encrypts the transmitted data by generating a random session key, thus protects the transmitted authentication information during the identification and authentication process.

RedCastle v3.0 for Asianux provides the management function, which is non-security function, of the system user and account of AXS3 through RedCastle Manager and the GUI for the Security Manager. The GUI is provided for convenience and the actual execution is done in AXS3 system.

RedCastle v3.0 for Asianux provides a non-security function to collect and store system logs generated in the AXS3 system. RedCastle Manager provides GUI with which the stored system logs can be referred to and searched. System log collection and storage function is used to store the system log separately and protect them by access control function, to enhance security of Asianux Server 3.

RedCastle v3.0 for Asianux provides a non-security function to establish and manage a policy for iptables function that performs network control. It also provides a function to collect, store, and refer to the log generated by iptables. Asianux Server 3 includes iptables v1.3.5.

RedCastle v3.0 for Asianux provides a non-security function to refer to system information and a report on the analysis of login history logs. RedCastle Manager provides GUI to be used for this function.

RedCastle Manager provides the security violation analysis report (security function) and the login analysis report (non-security function). It generates analysis and

statistics data on the audit data and calls the library provided as the IT environment to make a report. Runtime library and Java operational environment that are necessary for the report function to operate are provided at the installation of RedCastle Manager and installed with it.

- Java-based report library: JasperReports 2.0.1
- Java operational environment: Java 2 Runtime Environment, SE v1.4.2_03

3 TOE security environment

This chapter describes the assumptions describing the security of the TOE environment, the threats to the assets or environment of the TOE by the threat agent, and the organizational security policies, which are rules, procedures, practices, or guidelines to which the TOE should conform for the security.

3.1 Assumptions

The following headings are assumed to exist in the operational environment of the TOE that claims conformance to this ST.

A.Locate

The server on which the TOE is installed is located in a physically secure environment and protected from unauthorized physical modification.

A.Administrator

The administrator of the TOE is not malicious, is adequately trained, and correctly performs duties according to the administrator's guide.

A.OSpatch

The OS will be patched on its vulnerabilities and ensure no interference between the TOE and other applications.

A.Installation

The TOE is delivered, installed, generated, and operated in accordance with an appropriate procedure and ensured by the trusted administrator not to have an error in its security functions.

A.SSLprotocol

The SSL protocol, which RedCastle Manager implemented using openssl for the secure communication with RedCaste Agent, is secure.

A.Timestamp

Asianux Server 3 OS provides the TSF a reliable timestamp.

3.2 Threats

The IT assets to be protected include information that is transferred, processed, and stored by the TOE. The information means all data that exists in the TOE.

The assets can be defined as:

- TSF data
- User data of managed resources

The threat agent is assumed to have a little expertise, resources, and motivation, and can be defined as:

- Unauthorized TOE user: Not permitted to access the system
- Authorized TOE user: Permitted to access the system

3.2.1 Threats to the TOE

T.Masquerade

The threat agent may masquerade as an authorized user to obtain the authority.

T.Bypass

The threat agent may bypass the TOE security function and damage the TOE or an object.

T.Data

The threat agent may access the TOE data without authorization and affect the TOE security functions.

T.Integrity

The threat agent may take the TSF data being transmitted between RedCastle Manager and RedCastle Agent and corrupt it.

3.2.2 Threats to the IT environment

TE.Delivery

The person in charge of delivery and installation may corrupt the security of the TOE during the delivery and installation procedures.

3.3 Organizational security policy

The organizational security policies described in this section shall be observed by the TOE that claims conformance to this ST.

P.Audit

All security-relevant events shall be recorded and maintained and the data be reviewed to secure accountability of all security-relevant actions.

P.IA

An access to information shall be identified and authenticated before it is permitted.

P.Management

The authorized administrator shall manage the TOE in a secure manner.

P.Securitylevel

The TOE shall be able to assign a subject and object an appropriate security level or annul it in accordance with the access control policy and procedures of the organization.

P.Securityrole

The authorized administrator shall be able to manage and review the roles to establish and execute the role-based access control policy.

P.DAC

The TOE shall be able to control the access to information based on the identity of users or group to which they belong.

P.MAC

The TOE shall be able to control the access to information based on the security level of information and users.

P.RBAC

The TOE shall be able to control the access to an object based on the user's roles.

4 Security objectives

This ST defines the security objectives categorized into the security objectives for the TOE and for the environment. The former is addressed directly by the TOE and the latter is by the IT domain or non-technological/procedural means.

4.1 Security objectives for the TOE

This section describes the security objectives that shall be directly addressed by the TOE.

O.IA

The TOE shall uniquely identify its user and authenticate the user before permitting its access to ensure that only an authorized user may access the TOE.

O.Management

The TOE shall provide in a secure manner the authorized administrator with a means to manage the TOE effectively.

O.Audit

The TOE shall make and maintain the records on the security-relevant events so that all actions related to security may be accountable and provide a means to review the recorded data.

O.Data

The TOE shall protect TSF data stored in it from unauthorized exposure, modification, and deletion.

O.Securitylevel

The TOE shall be able to assign a subject and object an appropriate security level or annul it in accordance with the access control policy and procedures of the organization.

O.Securityrole

The TOE shall provide a means the authorized administrator to generate, delete, modify, and review user's roles.

O.DAC

The TOE shall be able to control the access to resources based on the identity of users or group to which they belong.

O.MAC

The TOE shall be able to control the access to information based on the security level of information and users.

O.RBAC

The TOE shall be able to control the access to an object based on the user's roles.

4.2 Security objectives for the environment

This section describes the security objectives that are to be met by the IT domain or non-technological/procedural means.

OE.Locate

The TOE is located in a physically secure environment where only an authorized administrator can access.

OE.Administrator

The administrator of the TOE is not malicious, is adequately trained, and correctly performs duties according to the administrator's guide.

OE.Securemanagement

The TOE shall be delivered and installed securely, and configured, managed, and used in a secure manner by an authorized administrator.

OE.OSpatch

The OS will be patched on its vulnerabilities by the TOE deleting useless services or means from the OS. The confidence and reliability of the OS shall be ensured.

OE.SSLprotocol

The TOE calls the SSL function provided by the IT environment to generate a secure communication channel between the Manager and Agent. The TOE performs authentication with the administrator’s ID and password and the access control list and shall protect the transmitted TSF data.

OE.Timestamp

The TOE shall record security-relevant events correctly by using the reliable timestamp provided in the IT environment.

5 IT security requirements

This chapter describes the security functional and assurance requirements that shall be satisfied by the TOE that claims conformance to this ST.

5.1 TOE security functional requirements

The TOE security functional requirements (SFR) in this ST are composed of the functional components from the CC Part 2. [Table 5-1] summarizes the functional components. This ST claims SOF-basic for the security functional requirements.

[Table 5-1] Security functional requirements

Security functional class	Security functional component	
Security audit	FAU_ARP.1	Security alarms
	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_SAA.1	Potential violation analysis
	FAU_SAR.1	Audit review
	FAU_SAR.2	Restricted audit review
	FAU_SAR.3	Selectable audit review
	FAU_SEL.1	Selective audit
	FAU_STG.1	Protected audit trail storage
	FAU_STG.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
User data protection	FDP_ACC.1(1)	Subset access control
	FDP_ACC.1(2)	Subset access control
	FDP_ACF.1(1)	Security attribute based access control
	FDP_ACF.1(2)	Security attribute based access control
	FDP_IFC.1	Subset information flow control
	FDP_IFF.2	Hierarchical security attributes
	FDP_ITC.1	Import of user data without security attributes
Identification and	FIA_AFL.1	Authentication failure handling
	FIA_ATD.1	User attribute definition

authentication	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	User authentication before any action
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.2	User identification before any action
	FIA_USB.1	User-subject binding
Security management	FMT_MOF.1	Management of security functions
	FMT_MSA.1(1)	Management of security attributes
	FMT_MSA.1(2)	Management of security attributes
	FMT_MSA.1(3)	Management of security attributes
	FMT_MSA.3(1)	Static attribute initialization
	FMT_MSA.3(2)	Static attribute initialization
	FMT_MSA.3(3)	Static attribute initialization
	FMT_MTD.1(1)	Management of TSF data
	FMT_MTD.1(2)	Management of TSF data
	FMT_MTD.1(3)	Management of TSF data
	FMT_MTD.1(4)	Management of TSF data
	FMT_REV.1(1)	Revocation
	FMT_REV.1(2)	Revocation
	FMT_SMF.1	Specification of management functions
	FMT_SMR.2	Restrictions on security roles
Protection of the TSF	FPT_AMT.1	Abstract machine testing
	FPT_RVM.1	Non-bypassability of the TSP
	FPT_TST.1	TSF testing
TOE access	FTA_LSA.1	Limitation on scope of selectable attributes
	FTA_MCS.1	Basic limitation on multiple concurrent sessions
	FTA_SSL.1	TSF-initiated session locking
	FTA_TSE.1	TOE session establishment

5.1.1 Security audit (FAU)

FAU_ARP.1 Security alarms

Hierarchical to: No other components

Dependencies: FAU_SAA.1 Potential violation analysis

FAU_ARP.1.1 The TSF shall take [the list below of the least disruptive actions] upon detection of a potential security violation.

a) [Compulsory kill of the violating process;

b) Alarm RedCastle Manager in real time; and

c) Inform the Security Manager by registered email of the potential security violation event]

FAU_GEN.1 Audit data generation

Hierarchical to: No other components

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1The 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 auditable events specified in [Table 5-2]]

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the ST, [the additional audit records in [Table 5-2]].

[Table 5-2] Auditable events

Functional component	Auditable event	Additional audit record
FAU_ARP.1	Actions taken due to imminent security violations	-
FAU_SAA.1	Enabling and disabling of any of the analysis	-

	mechanisms; Automated responses performed by the tool	
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating	-
FDP_ACF.1	Successful requests to perform an operation on an object covered by the SFP	Object identification information
FDP_IFF.2	All decisions on requests for information flow	Subject sensitivity label; Object sensitivity label
FDP_ITC.1	Successful import of user data, including any security attributes	-
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the action taken and the subsequent, if appropriate, restoration to the normal state	-
FIA_SOS.1	Rejection by the TSF of any tested secret	-
FIA_UAU.2	Unsuccessful use of the authentication mechanism	-
FIA_UID.2	Unsuccessful use of the user identification mechanism, including the user identity provided	-
FIA_USB.1	Unsuccessful binding of user security attributes to a subject	-
FMT_REV.1	All attempts to revoke security attributes	-
FMT_SMF.1	Use of the management functions	-
FMT_SMR.2	Modifications to the group of users that are part of a role	-
FTA_LSA.1	All failed attempts at selecting a session security attributes	
FTA_MCS.1	Rejection of a new session based on the limitation of multiple concurrent sessions	
FTA_SSL.1	Locking of an interactive session by the session locking mechanism; Successful unlocking of an interactive session	-
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism	

FAU_GEN.2 User identity association

Hierarchical to: No other components

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_SAA.1 Potential violation analysis

Hierarchical to: No other components

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of [identification and authentication failure of Agent, violation of access control rules];

b) [None]

FAU_SAR.1 Audit review

Hierarchical to: No other components

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAR.1.1 The TSF shall provide [the authorized administrator] with the capability to read [audit data generated in RedCastle Agent] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the **authorized administrator** to interpret the information.

FAU_SAR.2 Restricted audit review

Hierarchical to: No other components

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except **the authorized administrator** that has been granted explicit read-access.

FAU_SAR.3 Selectable audit review

Hierarchical to: No other components

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.3.1 The TSF shall provide the ability to perform *searches, sorting* of audit data based on [the following criteria].

a) [User identity;

b) Object identity;

c) Subject sensitivity label;

d) Object sensitivity label;

e) Audit period; and

f) Level of warning]

FAU_SEL.1 Selective audit

Hierarchical to: No other components

Dependencies: FAU_GEN.1 Audit data generation

FMT_MTD.1 Management of TSF data

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) Object identity, user identity, subject identity, event type

b) [Subject sensitivity label, object sensitivity label, audit period, level of warning]

FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1 The TSF shall protect the stored audit records from unauthorized deletion.

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

FAU_STG.3 Action in case of possible audit data loss

Hierarchical to: No other components

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.3.1 The TSF shall take [the following actions] if the audit trail exceeds [80% of the audit storage].

a) [Alarm RedCastle Manager and inform the administrator by registered email if the audit trail exceeds 80% of audit storage;

b) Alarm RedCastle Manager and inform the administrator by registered email at every increase by 5% (at 85%, 90%, 95% of the audit storage); and

c) If the audit trail reaches 100%, alarm RedCastle Manager, inform the administrator by registered email, and perform the functions of FAU_STG.4, Prevention of audit data loss].

FAU_STG.4 Prevention of audit data loss

Hierarchical to: FAU_STG.3

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.4.1 The TSF shall *overwrite the oldest stored audit records* and [none] if the audit trail is full.

5.1.2 User data protection (FDP)

FDP_ACC.1(1) Subset access control

Hierarchical to: No other components

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the [DAC policy] on [the following list of subjects, objects, and operations among subjects and objects covered by the SFP].

a) [List of subjects: user, process

b) List of objects: command and process of the OS

c) List of operations among subjects and objects:

   i) Execute

   ii) Kill]

FDP_ACC.1(2) Subset access control

Hierarchical to: No other components

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the [RBAC policy] on [the following list of subjects, objects, and operations among subjects and objects covered by the SFP].

a) [List of subjects: process operating on behalf of a user

b) List of objects: directory and file of the OS

c) List of operations among subjects and objects

   i) read,

ii) write,
iii) execute,
iv) create,
v) delete,
vi) rename,
vii) chmod,
viii) chown]

FDP_ACF.1(1) Security attribute based access control

Hierarchical to: No other components

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1 The TSF shall enforce the [DAC policy] to objects based on the following:

a) [List of subjects: user identity, process role status

b) List of objects: execution file, process]

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) [List of command control: If a subject the process role of which is UXR accesses an execution file registered on the command control list in the status of the execution operation, deny the execution.

b) List of Kill prevention process: If a subject of which the process roles are UXR and SA accesses a process registered on the Kill prevention process list in the status of unallowed Kill operation, deny the enforced disabling of the process.

c) List of SETUID execution permission: If a user accesses the SETUID file registered on the list in the status of execution operation, allow the access except for the user whose identity is root.

d) List of permitted su user: The operation is allowed if the privilege moves from root to a user registered on the list using su. Otherwise deny it.]

FDP_ACF.1.3 The TSF shall explicitly authorize 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 rules].

FDP_ACF.1(2) Security attribute based access control

Hierarchical to: No other components

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

FDP_ACF.1.1 The TSF shall enforce the [RBAC policy] to objects based on the following:

a) [List of subjects: user role, user identity, user group identity, security group

b) Additional list of subjects: process executed by a user or subject process, user IP address

c) List of objects: file]

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) [If an operation enforced by a subject is explicitly allowed in the access control policy to objects, enforce the RBAC policy.]

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

a) [If the security objective assigned to a subject is consistent with that specified in the policy, explicitly authorize access of the subject to the object.

b) Comparing the security attributes assigned to a subject will consider the user role, user identity, user group identity, security group, process executed by a subject, and user IP address in that order.]

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [following rule].

a) [If the security attribute of a subject is not consistent with that described in the policy, explicitly deny access of the subject to the object.]

FDP_IFC.1 Subset information flow control

Hierarchical to: No other components

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the [MAC policy] on [the following list of subjects, objects, operations **between subjects**, and operations **between subjects and objects**].

a) [List of subjects: user, process executed by a user

b) List of objects:

   i) Between subjects: the subject upon which operation is performed is the user and process in the LINUX OS.
   ii) Between subjects and objects: the object upon which operation is performed is the user, process, and file in the LINUX OS.

c) List of operations between subjects

   i) READ operation: kill

d) List of operations between subjects and objects

   i) READ operation: read, execute
   ii) WRITE operation: write, delete, generate]

FDP_IFF.2 Hierarchical security attributes

Hierarchical to: FDP_IFF.1

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialization

FDP_IFF.2.1 The TSF shall enforce the [MAC policy] based on the following types of subject and information security attributes:

a) [Subject sensitivity label: security level, security category, process role status

b) Object sensitivity label: security level, security category

c) Security level: 6 scales of protection

- Top Secret (1)
- Secret (2)
- Confidential (3)
- Restricted (4)
- Unclassified (5-7)
- None (0)

d) Security category: 0 ~ 127

- Security officer (SO): The highest category with the Category ID 1
- System administrator (SA): Sub-category with the Category ID 2 and below
- Multi-label security user (MU): Sub-category with the Category ID below 1 and excluded from below 2
- System user (UX): Category ID is 0; root is always assigned to this category

e) Process role status (RS): Additional security attribute that is assigned to a process operating on behalf of a user; automatically decided as one of 7 scales according to the security category and process role.

- SO, MSO status: When the user is in the SO category; SO's process role is root; MSO's process role is the user. (uid is not 0)
- SA, MSA status: When the user is in the SA category; SA's process role is root; MSA's process role is the user. (uid is not 0)
- MU status: When the user is in the MU category.
- UXR, UX status: User is in the UX category; UXR's process role is root; UX's process role is the user. (uid is not 0)]

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules, based on the ordering relationships between security attributes, hold:

a) [A subject obtains read access right to an object if the subject sensitivity is

same as or higher than the object sensitivity.

b) A subject obtains write access right to an object if the object sensitivity is same as the subject sensitivity.

c) Information may flow from subject B to subject A if the sensitivity of subject A is same as or higher than that of subject B.]

FDP_IFF.2.3 The TSF shall enforce the [None].

FDP_IFF.2.4 The TSF shall provide the following [None].

FDP_IFF.2.5 The TSF shall explicitly authorize an information flow based on the following rules:

a) [When an execution file including the path registered in the attribute re-establish list is being executed after allowing execution, always allow the execution if the role status of process is SA, UXR.

b) After allowing execution, re-establish the security level and category of process in accordance with the value specified in the list.]

FDP_IFF.2.6 The TSF shall explicitly deny an information flow based on the following rules: [None].

FDP_IFF.2.7 The TSF shall enforce the following relationships for any two valid **sensitivity label**:

a) There exists an ordering function that, given two valid security attributes, determines:

   i) if the **sensitivity labels** are equal;
   ii) if one **sensitivity label** is greater than the other; and
   iii) if the **sensitivity labels** are incomparable.

b) There exists a “least upper bound (LUB)” in the set of **sensitivity labels**, such that, given any two valid **sensitivity labels**, there is a valid **sensitivity label** that is greater than or equal to the two valid **sensitivity labels**; and

c) There exists a “greatest lower bound (GLB) in the set of **sensitivity labels**, such that, given any two valid **sensitivity labels**, there is a valid

sensitivity label that is not greater than the two valid **sensitivity labels**.

FDP_ITC.1 Import of user data without security attributes

Hierarchical to: No other components

Dependencies: [FDP_IFC.1 Subset information flow control]

FMT_MSA.3 Static attribute initialization

FDP_ITC.1.1 The TSF shall enforce the [MAC policy] when importing user data, controlled under the **MAC policy**, from outside of the TSC.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the **MAC policy** from outside the TSC:

a) [An authorized administrator shall specify the sensitivity label of the user data imported from outside the TSC based on the subject that imports data.

b) An authorized administrator shall specify the sensitivity label of the user data imported from outside the TSC based on the sensitivity label of the subject that imports data.]

5.1.3 Identification and authentication (FIA)

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 The TSF shall detect when *[5]* unsuccessful authentication attempts occur related to [the administrator's Agent authentication].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [stop authenticating the user until the Security Manager takes action and inform the Security Manager by registered email.]

FIA_ATD.1 User attribute definition

Hierarchical to: No other components

Dependencies: No dependencies

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) [User identity

b) User group identity

c) Security label: Security level, security category

d) Authentication data]

FIA_SOS.1 Verification of secrets

Hierarchical to: No other components

Dependencies: No dependencies

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [the following defined quality metric].

a) [Password length: 8 through 15 characters

b) Acceptable password characters are:

   i) 52 alphabetic letters (lowercase or uppercase)
   ii) 10 numeric digits (0-9)
   iii) 16 special characters (!, @, #, $, %, ^, &, *, (,), +, =, <, >, :, ;)

c) Password must contain at least one alphabetic letter, numeric digit, and special character.

d) Consecutive numeric or alphabetic characters are allowed.

e) Repeated use of same alphabetic, numeric, and special characters are allowed.]

Application notes: Examples of the defined quality metric for a password authentication mechanism may include minimum length, combination rule, or change period.

FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UAU.7.1 The TSF shall provide only ['*' or blank] to the user while the authentication is in progress.

FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1

Dependencies: No dependencies

FIA_UID.2.1 The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

FIA_USB.1 User-subject binding

Hierarchical to: No other components

Dependencies: FIA_ATD.1 User attribute definition

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

a) [User identity

b) User group identity

c) Security level, security category

d) Process role status]

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on behalf of users:

a) [If a user is successfully identified and authenticated, the security level and category given to the user and the process role determined by the category are assigned to the subject process.]

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on behalf of users:

a) [User identity and group identity may change by su command. When a user is successfully identified and authenticated using su, the security level, security category, and process role determined by the category will be given to the newly generated subject process according to its identity. The security attribute of the process before executing su, however, will not be changed.

b) When a program registered in the attribute re-establish list is being executed after allowing execution, re-establish the security level and category of process in accordance with the value specified in the list.]

5.1.4 Security management (FMT)

FMT_MOF.1 Management of security functions behavior

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of management functions

FMT_MOF.1.1 The TSF shall restrict the ability to *disable, enable* the functions [described below] to [the authorized administrator].

a) [RedCastle Agent communication server

b) Audit data collection function

c) RedCastle kernel module

d) RBAC function

e) SETUID execution control function

f) Kill control function

g) Login service control function]

FMT_MSA.1(1) Management of security attributes

Hierarchical to: No other components

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

FMT_MSA.1.1 The TSF shall enforce the [DAC policy] to restrict the ability to *change_default, query, modify* the security attributes [DAC associated with a user, execution file, and process] to [the authorized administrator].

FMT_MSA.1(2) Management of security attributes

Hierarchical to: No other components

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

FMT_MSA.1.1 The TSF shall enforce the [RBAC policy] to restrict the ability to *change_default, query, modify* the security attributes [RBAC associated with a user, user role, and process] to [the authorized administrator].

FMT_MSA.1(3) Management of security attributes

Hierarchical to: No other components

Dependencies: [FDP_ACC.1 Subset access control, or

FDP_IFC.1 Subset information flow control]

FMT_SMF.1 Specification of management functions

FMT_SMR.1 Security roles

FMT_MSA.1.1 The TSF shall enforce [MAC policy] to restrict the ability to *change_default, query, modify* the security attributes [MAC associated with a user, process being executed on behalf of a user, and file] to [the authorized administrator].

FMT_MSA.3(1) Static attribute initialization

Hierarchical to: No other components

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the [DAC policy] to provide *restrictive* default values for security attributes that are used to enforce the **DAC policy**.

FMT_MSA.3.2 The TSF shall allow the [authorized administrator] to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3(2) Static attribute initialization

Hierarchical to: No other components

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the [RBAC policy] to provide *restrictive* default values for security attributes that are used to enforce the **RBAC policy**.

FMT_MSA.3.2 The TSF shall allow the [authorized administrator] to specify alternative initial values to override the default values when an object or information is created.

FMT_MSA.3(3) Static attribute initialization

Hierarchical to: No other components

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

FMT_MSA.3.1 The TSF shall enforce the [MAC policy] to provide *restrictive* default values for security attributes that are used to enforce the **MAC policy**.

FMT_MSA.3.2 The TSF shall allow the [authorized administrator] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1(1) Management of TSF data

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of management functions

FMT_MTD.1.1 The TSF shall restrict the ability to *query* the [audit data] to [the authorized administrator].

FMT_MTD.1(2) Management of TSF data

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of management functions

FMT_MTD.1.1 The TSF shall restrict the ability to *delete, [initialize]* the [identification and authentication data] to [the authorized administrator].

FMT_MTD.1(3) Management of TSF data

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of management functions

FMT_MTD.1.1 The TSF shall restrict the ability to *modify* the [authentication data] to [the authorized administrator and the user authorized to modify its own authentication data].

FMT_MTD.1(4) Management of TSF data

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of management functions

FMT_MTD.1.1 The TSF shall restrict the ability to *change_default, query, delete, clear, [generate]* the [TSF data associated with security other than those mentioned above] to [the authorized administrator].

FMT_REV.1(1) Revocation

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the *users* within the TSC to [the authorized administrator].

FMT_REV.1.2 The TSF shall enforce the rules:

a) [Revocation of the security attributes of users must done at the time of the deletion of users.]

FMT_REV.1(2) Revocation

Hierarchical to: No other components

Dependencies: FMT_SMR.1 Security roles

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the *objects* within the TSC to [the authorized administrator].

FMT_REV.1.2 The TSF shall enforce the rules:

a) [Revocation of the security attributes of objects must done at the time of the deletion of objects.]

FMT_SMF.1 Specification of management functions

Hierarchical to: No other components

Dependencies: No dependencies

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

[List of security management functions to be provided by the TSF:

a) Startup or stop of security functions

b) Management of security groups

c) Management of MU

d) Management of user roles

e) Management of the security attributes of objects

f) Management of the security attributes of subjects

g) Management of role-based policy

h) Management of allow/deny list

i) Configuration of audit data

j) Management of Manager users

k) Management of Agent authentication data]

FMT_SMR.2 Restrictions on security roles

Hierarchical to: FMT_SMR.1

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.2.1 The TSF shall maintain the roles:

a) [Authorized administrator

b) User authorized to modify its own authentication data

c) System user (UX) that cannot modify authentication data]

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions [the following conditions regarding generation of user roles and assignment of users to those roles] are satisfied.

a) [Different user roles cannot have hierarchical structure.

b) The maximum number of users that can be assigned to a user role is 8.

c) The maximum number of groups that can be assigned to a user role is 8.

d) The number of security level that can be assigned to a user role is 1.

e) The number of security category that can be assigned to a user role is 1, excluding the sub-categories.]

5.1.5 Protection of the TSF (FPT)

FPT_AMT.1 Abstract machine testing

Hierarchical to: No other components

Dependencies: No dependencies

FPT_AMT.1.1 The TSF shall run a suite of tests *during initial start-up, periodically during normal operation, at the request of an authorized* ***administrator*** to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components

Dependencies: No dependencies

FPT_RVM.1.1The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_TST.1 TSF testing

Hierarchical to: No other components

Dependencies: FPT_AMT.1 Abstract machine testing

FPT_TST.1.1 The TSF shall run a suite of self tests during initial start-up, periodically during normal operation, at the request of the authorized **administrator** to demonstrate the correct operation of the TSF.

FPT_TST.1.2 The TSF shall provide authorized **administrators** with the capability to verify the integrity of *TSF data*.

FPT_TST.1.3 The TSF shall provide authorized **administrators** with the capability to verify the integrity of stored TSF executable code.

5.1.6 TOE access (FTA)

FTA_LSA.1 Limitation on scope of selectable attributes

Hierarchical to: No other components

Dependencies: No dependencies

FTA_LSA.1.1 The TSF shall restrict the scope of the session security attributes [session service name, access IP address, range of access time, allowance and denial of access], based on [user identity or group identity and all users].

FTA_MCS.1 Basic limitation on multiple concurrent sessions

Hierarchical to: No other components

Dependencies: FIA_UID.1 Timing of identification

FIA_MCS.1.1 The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.

FIA_MCS.1.2 The TSF shall enforce, by default, a limit of [99] sessions per user.

FTA_SSL.1 TSF-initiated session locking

Hierarchical to: No other components

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL.1.1 The TSF shall lock an interactive session after [time interval of the authorized administrator inactivity exceeds the wait-time of screen saver] by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL1.2 The TSF shall require the following events to occur prior to unlocking the session: [authentication of the authorized administrator in the system].

FTA_TSE.1 TOE session establishment

Hierarchical to: No other components

Dependencies: No dependencies

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on:

a) [Service policy by user and group: Service, access IP address, and access time range that will be denied

b) Service policy for all users: Service, access IP address, and access time range that will be denied

c) Denied IP address for all users

d) Session control by user and group: Maximum number of acceptable concurrent sessions

e) Session control for all users: Maximum number of acceptable concurrent sessions]

5.2 TOE security assurance requirements

The TOE security assurance requirements (SAR) in this ST are composed of the assurance components from the CC Part 3. The targeted assurance level in this ST is EAL4. [Table 5-3] summarizes the assurance components this ST requires.

[Table 5-3] Security assurance requirements

Assurance class	Assurance component	
Configuration management	ACM_AUT.1	Partial CM automation
	ACM_CAP.4	Generation support and acceptance procedures
	ACM_SCP.2	Problem tracking CM coverage
Delivery and operation	ADO_DEL.2	Detection of modification
	ADO_IGS.1	Installation, generation, and start-up procedures
Development	ADV_FSP.2	Fully defined external interfaces
	ADV_HLD.2	Security enforcing high-level design
	ADV_IMP.1	Subset of the implementation of the TSF
	ADV_LLD.1	Descriptive low-level design
	ADV_RCR.1	Informal correspondence demonstration
	ADV_SPM.1	Informal TOE security policy model
Guidance documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Life cycle support	ALC_DVS.1	Identification of security measures
	ALC_LCD.1	Developer defined life-cycle model
	ALC_TAT.1	Well-defined development tools
Tests	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: high-level design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing – sample
Vulnerability assessment	AVA_MSU.2	Validation of analysis
	AVA_SOF.1	Strength of TOE security function evaluation TOE
	AVA_VLA.2	Independent vulnerability analysis

5.2.1 Configuration management (ACM)

ACM_AUT.1 Partial CM automation

Dependencies: ACM_CAP.3 Authorization controls

Developer action elements:

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

Content and presentation of evidence elements:

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

Evaluator action elements:

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

ACM_CAP.4 Generation support and acceptance procedures

Dependencies: ALC_DVS.1 Identification of security measures

Developer action elements:

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C The TOE shall be labeled with its reference.

ACM_CAP.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

ACM_CAP.4.4C The configuration list shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.5C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.6C The CM documentation shall describe the method used to uniquely identify the configuration items that comprises the TOE.

ACM_CAP.4.7C The CM system shall uniquely identify all configuration items that comprise the TOE.

ACM_CAP.4.8C The CM plan shall describe how the CM system is used.

ACM_CAP.4.9C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.10C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.11C The CM system shall provide measures such that only authorized changes are made to the configuration items.

ACM_CAP.4.12C The CM system shall support the generation of the TOE.

ACM_CAP.4.13C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

Evaluator action elements:

ACM_CAP.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ACM_SCP.2 Problem tracking CM coverage

Dependencies: ACM_CAP.3 Authorization controls

Developer action elements:

ACM_SCP.2.1D The developer shall provide a list of configuration items for the TOE.

Content and presentation of evidence elements:

ACM_SCP.2.1C The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.

Evaluator action elements:

ACM_SCP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2 Delivery and operation (ADO)

ADO_DEL.2 Detection of modification

Dependencies: ACM_CAP.3 Authorization controls

Developer action elements:

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D The developer shall use the delivery procedures.

Content and presentation of evidence elements:

ADO_DEL.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Evaluator action elements:

ADO_DEL.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1 Installation, generation, and start-up procedures

Dependencies: AGD_ADM.1 Administrator guidance

Developer action elements:

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

ADO_IGS.1.1C The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.

Evaluator action elements:

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

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.2.3 Development (ADV)

ADV_FSP.2 Fully defined external interfaces

Dependencies: ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_FSP.2.1D The developer shall provide a functional specification.

Content and presentation of evidence elements:

ADV_FSP.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.2.2C The functional specification shall be internally consistent.

ADV_FSP.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.2.4C The functional specification shall completely represent the TSF.

ADV_FSP.2.5C The functional specification shall include rationale that the TSF is completely represented.

Evaluator action elements:

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.2 Security enforcing high-level design

Dependencies: ADV_FSP.1 Informal functional specification

ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements:

ADV_HLD.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_IMP.1 Subset of the implementation of the TSF

Dependencies: ADV_LLD.1 Descriptive low-level design

ADV_RCR.1 Informal correspondence demonstration

ALC_TAT.1 Well-defined development tools

Developer action elements:

ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF.

Content and presentation of evidence elements:

ADV_IMP.1.1C The implementation representation shall unambiguously 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 internally consistent.

Evaluator action elements:

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

ADV_IMP.1.2E The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

ADV_LLD.1 Descriptive low-level design

Dependencies: ADV_HLD.2 Security enforcing high-level design

ADV_RCR.1 Informal correspondence demonstration

Developer action elements:

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

Content and presentation of evidence elements:

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

ADV_LLD.1.2C The low-level design shall be internally consistent.

ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.1.4C The low-level design shall describe the purpose of each module.

ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

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

ADV_LLD.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.1 Informal correspondence demonstration

Dependencies: No dependencies

Developer action elements:

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements:

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

ADV_SPM.1 Informal TOE security policy model

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

Content and presentation of evidence elements:

ADV_SPM.1.1C The TSP model shall be formal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

Evaluator action elements:

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

5.2.4 Guidance documents (AGD)

AGD_ADM.1 Administrator guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

Content and presentation of evidence elements:

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

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

AGD_USR.1 User guidance

Dependencies: ADV_FSP.1 Informal functional specification

Developer action elements:

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements:

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

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

5.2.5 Life cycle support (ALC)

ALC_DVS.1 Identification of security measures

Dependencies: No dependencies

Developer action elements:

ALC_DVS.1.1D The developer shall produce development security documentation.

Content and presentation of evidence elements:

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.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

Evaluator action elements:

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_LCD.1 Developer defined life-cycle model

Dependencies: No dependencies

Developer action elements:

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.

Content and presentation of evidence elements:

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.

Evaluator action elements:

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

Dependencies: ADV_IMP.1 Subset of the implementation of the TSF

Developer action elements:

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

Content and presentation of evidence elements:

ALC_TAT.1.1C All development tools used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

Evaluator action elements:

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

5.2.6 Tests (ATE)

ATE_COV.2 Analysis of coverage

Dependencies: ADV_FSP.1 Informal functional specification

ATE_FUN.1 Functional testing

Developer action elements:

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

Content and presentation of evidence elements:

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

Evaluator action elements:

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_DPT.1 Testing: high-level design

Dependencies: ADV_HLD.1 Descriptive high-level design

ATE_FUN.1 Functional testing

Developer action elements:

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements:

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

Evaluator action elements:

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

ATE_FUN.1 Functional testing

Dependencies: No dependencies

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements:

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

Dependencies: ADV_FSP.1 Informal functional specification

AGD_ADM.1 Administrator guidance

AGD_USR.1 User guidance

ATE_FUN.1 Functional testing

Developer action elements:

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

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.

Evaluator action elements:

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 test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.2.7 Vulnerability assessment (AVA)

AVA_MSU.2 Validation of analysis

Dependencies: ADO_IGS.1 Installation, generation, and start-up procedures

ADV_FSP.1 Informal functional specification

AGD_ADM.1 Administrator guidance

AGD_USR.1 User guidance

Developer action elements:

AVA_MSU.2.1D The developer shall provide guidance documentation.

AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

Content and presentation of evidence elements:

AVA_MSU.2.1C The guidance documentation 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.

AVA_MSU.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

Evaluator action elements:

AVA_MSU.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.2.2E The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA_SOF.1 Strength of TOE security function evaluation

Dependencies: ADV_FSP.1 Informal functional specification

ADV_HLD.1 Descriptive high-level design

Developer action elements:

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

Content and presentation of evidence elements:

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:

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

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

AVA_VLA.2 Independent vulnerability analysis

Dependencies: ADV_FSP.1 Informal functional specification

ADV_HLD.2 Security enforcing high-level design

ADV_IMP.1 Subset of the implementation of the TSF

ADV_LLD.1 Descriptive low-level design

AGD_ADM.1 Administrator guidance

AGD_USR.1 User guidance

Developer action elements:

AVA_VLA.2.1D The developer shall perform a vulnerability analysis.

AVA_VLA.2.2D The developer shall provide vulnerability analysis documentation.

Content and presentation of evidence elements:

AVA_VLA.2.1C The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

AVA_VLA.2.2C The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

AVA_VLA.2.3C The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.2.4C The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

Evaluator action elements:

AVA_VLA.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.2.2E The evaluator ***shall conduct*** penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.2.3E The evaluator ***shall perform*** an independent vulnerability analysis.

AVA_VLA.2.4E The evaluator ***shall perform*** independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.2.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low attack potential.

5.3 Security requirements for the IT environment

The security requirements for the IT environment are:

Functional class	Functional component	
Protection of the TSF	FPT_ITT.1	Basic internal TSF data transfer protection
	FPT_STM.1	Reliable time stamps

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components

Dependencies: No dependencies

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

FPT_STM.1 Reliable time stamps

Hierarchical to: No other components

Dependencies: No dependencies

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

5.4 Strength of function

This ST defines the minimum strength of function level as SOF-basic against attackers possessing a low expertise, resources, and motivation. Therefore, the strength of function required for this ST is SOF-basic, which satisfies the quality metric for verifying secrets specified in FIA_SOS.1 and applies for FIA_UAU.2.

6 TOE summary specification

6.1 TOE security functions

6.1.1 Security audit (AU)

6.1.1.1 Audit data generation and collection (AU.1)

The following audit data are generated in the TOE and are collected and stored in Log Daemon.

- Security log generated by the security management function
- Kernel log generated by RedCastle kernel module

The Log Daemon provides a function, which is categorized into non-security functions of the TOE, to collect iptables log, system log, and those listed below from outside the TOE and store them in a specified place.

- Log generated in iptables
- List of currently logged-in user
- Login failure log
- User login history
- Most recent user login information
- Syslog: messages

The security management function of the TOE and kernel module generates an audit log on the following auditable events (to satisfy the requirement FAU_GEN.1):

- Action taken upon detection of potential security violation
  - Compulsory shut-down of subject process
  - Emailing the administrator
- Start-up and shut-down of the audit functions
- Changes to the security audit configuration
- Backup and initialization of an audit log

- Operation control of DAC function
- Operation control of RBAC function
- Operation control of MAC function
- Successful and failed identification and authentication of Agent
- The reaching of the threshold for the unsuccessful authentication attempts and the actions
- Changes to a security group and MU attributes
- Changes to user roles
- Changes to the security attributes of an object
- Communication session errors between Manager and Agent
- Detection of accumulated occurrence of access control violation log and actions
- Warning about and actions to a full audit trail
- Blocking insecure attempts to access Manager
- Addition, modification, and deletion of RBAC policy
- Addition, modification, and deletion of MAC policy
- Addition, modification, and deletion of allow/deny list
- Configuration of security functions
- Management of a list of the items to be checked for an integrity and the integrity check
- Start-up and stop of an access control security function
- Addition, modification, and deletion of user service policy
- Control of a user service
- Modification of a user session
- Control of a user session
- Successful and failed identification and authentication of Manager
- Registration and deletion of a Manager user and password change

- Setting of Manager display locking
- Manager display locking
- Successful and failed unlocking of Manager display

The TOE provides a selective audit function, with which the SO can generate an audit log only for the necessary auditable events by configuration of the security environment (to satisfy the requirements FAU_SEL.1).

Each security log comprises audit data including the following articles:

- Time of occurrence of an audit
- Place of audit: Communication daemon, log daemon, and kernel module, which generate audit data
- Level of warning
  - Information: For normal events such as successful modification of a policy by the SO, start-up and stop of a security function, etc.
  - Notice: Security violation events such as authentication failure, access control violation, etc.
  - Warning: Events categorized into potential violations due to the accumulation of Notice security violation events
  - Critical: Events occurred due to, for example, audit storage exhaustion
  - Error: Failed attempt of the SO to change a policy
- Audit message

The access control violation log of RedCastle kernel module among auditable security logs comprises audit data for the following additional articles and is stored in a security log file (to satisfy the requirement FAU_GEN.2):

- Subject information
  - Effective User ID
  - Process ID
  - Process security group
  - Process security level

  - Process role status
- Object information
  - File and directory name
  - Object security group
  - Object security level
- Occurred operation: System call information
- Violation message

Time of occurrence of an event means a reliable timestamp that the TSF intends to use. The TOE uses the timestamp provided by Asianux Server 3, the IT environment of the TOE.

The audit data for reference monitor, MAC, DAC, and RBAC functions are generated on the kernel level, while those for identification, authentication, security management, and protection of the TSF are on the application level. The Agent log daemon provides a non-security function that collects syslog generated in the system and stores it in the place same as the security log as a separate file.

SFRs to be met: FAU_GEN.1, FAU_GEN.2, FAU_SEL.1

6.1.1.2 Potential violation analysis and action (AU.2)

The SO of the TOE may define a unit time and a limit of accumulated number of violation events per unit time for the analysis of potential violation on collected audit data. If the number of security violation reviewed among the collected audit data exceeds the limit number, the TSF detects it and gives warning (to satisfy the security requirement FAU_SAA.1).

The following articles may be defined in the TOE for analysis of potential security violation.

- Unit time for potential security violation analysis
- The limit number for each security violation

The set of rules for potential security violation analysis includes:

- Violation of an identification and authentication security policy
- Violation of an access control rule

The TOE provides functions to analyze and act to potential violation attacks (to satisfy the security requirement FAU_ARP.1) for the cases as:

- Violation of an identification and authentication security policy: In the case that an identification and authentication attempt consecutively fails 5 times in Agent identification and authentication (IA.2), stop the authentication, alarm RedCastle Manager in real time and inform the SO by email.
- Violation of an access control rule: If a series of security violations, among collected access control violation audit data, done by a user exceeds the limit number of accumulation per unit time, KILL the process and notify RedCastle Manager in real time and the SO by email.

SFRs to be met: FAU_ARP.1, FAU_SAA.1

6.1.1.3 Audit storage management (AU.3)

The TOE enforces LBAC on the files and directories that store audit records to protect them from unauthorized deletion or modification and prevent users except for the SO from reading them. To this end, the TOE assigns SO group and Top Secret level of security to the files and directories (to satisfy the security requirements FAU_SAR.2 and FAU_STG.1).

The TOE provides the file size of 10 MB and number of 5 as the default audit storage and takes action for protection of the audit trail as below (to satisfy the security requirement FAU_STG.3):

a) Alarm Manager and inform the administrator by registered email if the audit trail exceeds 80% of audit storage;

b) Alarm Manager and inform the administrator by registered email at every increase by 5% (at 85%, 90%, 95% of the audit storage); and

c) If the audit trail reaches 100%, alarm Manager, inform the administrator by

registered email, and perform the function of prevention of audit data loss.

The case of c) above means the exhaustion of audit storage, in which case the TOE performs the following functions to prevent audit data loss (to satisfy the security requirement FAU_STG.4):

a) Overwrite the oldest stored audit records: If there are 5 audit files of 10 MB to make 50M of audit records, replace the oldest audit file with the second oldest one and store it in rclog.01 file.

SFRs to be met: FAU_SAR.2, FAU_STG.1, FAU_STG.3, FAU_STG.4

6.1.1.4 Audit data reference and review (AU.4)

The administrator can refer to and review the security logs generated by RedCastle Agent through the GUI provided by RedCastle Manager (to satisfy the security requirement FAU_SAR.1).

The TOE provides a function to search and sort audit logs by categories as below (to satisfy the security requirement FAU_SAR.3):

- User identity
- Object identity
- Subject sensitivity label
- Object sensitivity label
- Time of occurrence of an audit
- Place of audit
- Level of warning

The TOE can generate a report on security violation analysis of the audit data generated by RedCastle Agent through the GUI provided by RedCastle Manager.

The TOE provides a non-security function to refer to and review the system log and iptables log and also provides, based on the system log, login analysis report by a user and an IP.

SFRs to be met: FAU_SAR.1, FAU_SAR.3

6.1.2 Identification and authentication (IA)

Identification and authentication functions of the TOE include identification and authentication for using Manager (IA.1, Manager identification and authentication) and identification and authentication of the Security Manager in accessing Agent through Manager and Java Manager (IA.2, Agent identification and authentication).

6.1.2.1 Manager identification and authentication (IA.1)

The Manager administrator and user (“Manager User”) can access RedCastle Agent after the Manager identification and authentication of RedCastle Manager.

Manager User should, through the GUI, input the following values (to satisfy the security requirements FIA_UAU.2 and FIA_UID.2):

a) Manager user ID

b) Manager user password

The identification and authentication process of RedCastle Manager will follow the steps below (to satisfy the security requirement FIA_UAU.7):

a) User inputs id and password through the GUI.

b) Password input will be shown ‘*’ to protect authentication feedback.

c) Decrypts the authentication data encrypted in DB based on the ID and password.

d) Check if there is a match among the decrypted authentication data to the user ID.

e) If there is a match, it means that the input ID and password are verified and the process is complete.

If the number of failed authentication attempts exceeds the limit (default: 5), compulsorily shut down RedCastle Manager to prevent possible repetitive attempts.

SFRs to be met: FIA_UAU.2, FIA_UAU.7, FIA_UID.2

6.1.2.2 Agent identification and authentication (IA.2)

After successful login to RedCastle Manager, the Manager User can access each server on which RedCastle Agent is installed. Each server requires identification and authentication of administrator for its Agent (to satisfy the security requirements FIA_UAU.2 and FIA_UID.2). The Security Manager can perform identification and authentication process through the GUI of Java Manager after login to X Windows from the system console. Java Manager only allows access from local address (127.0.0.1) and does not provide access to remote Agent. The identification and authentication information that the SO provides through the GUI for this function include:

- System account (SO ID)
- System account password
- Security password

System account and its password refer to the account of the OS of the system on which RedCastle Agent is installed. Security password is chosen at the register of SO, which is separately treated from the system account. For the protection of feedback, password is displayed '*' while being input (to satisfy the security requirement FIA_UAU.7).

If the authentication attempt of a user fails 5 times consecutively, disconnect the access to RedCastle Agent and prevent further attempt until the SO remove the prevention of that user. Make a report on the event of prevention of authentication and email at the registered address. A SO that is once prevented from being authenticated can only be released by another SO removes the prevention through the GUI of Manager (to satisfy the security requirement FIA_AFL.1).

For the management of MU, only a user given the right of SO may log in Agent through Manager, then a communication session for processing command of security functions and another for real-time searching for a log are assigned to the SO. The access types between the Manager and Agent categorize into:

- Management mode: User can perform security management like establishing security policies and all security functions including log reference.
- Monitoring mode: Only the functions of real-time log reference and log search are provided.

Management mode allows only one access while Monitoring mode allows many. Only those assigned the attribute of (SO, Top Secret) can access Management mode; those assigned SO but not Top Secret can only access Monitoring mode. A user with (SO, Top Secret) has the role of Security Manager. If the SO attempts to access Manager when the same account as the SO already has an access to the Management mode session, the SO can block the existing session enforcedly and access; if there is a Management mode session on from other account, the SO can access the Monitoring mode. A user that falls under the SO group and with the level of other than Top Secret is the General Manager, who can be assigned to the Monitoring mode after accessing Agent through Manager (to satisfy the security requirement FIA_UID.2).

The communication between RedCastle Manager and RedCastle Agent is done by SSL version 3 protocol provided in the IT environment. The key exchange and authentication method is anonymous authentication mechanism (anonymous DH); AES cryptographic algorithm (256 bit encryption key) and SHA hash algorithm are selected for the cipher-suite (ADH-AES256-SHA). The SSL protocol encrypts the transmitted data by generating a random session key, thus protects the transmitted authentication information during the identification and authentication process.

Agent identification and authentication employs a password authentication mechanism that requires statistical analysis, thus requiring analysis of SOF. It satisfies SOF-Basic to counter a threat agent possessing a low attack potential.

SFRs to be met: FIA_AFL.1, FIA_UAU.2, FIA_UAU.7, FIA_UID.2

6.1.3 Access control (AC)

6.1.3.1 Reference monitor (AC.1)

The TSF provides a reference monitor to ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed. The reference monitor function is provided in the TOE for the following purpose:

- To ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.
- A system call is added to be used in the TOE for the establishment of security policies and environment of RedCastle Agent. It will also be used in collecting audit records.

The reference monitor performs the processes of adding a system call when the kernel module of RedCastle is loaded into the OS kernel at the start-up of OS and of exchanging the system call of OS when the security function starts to operate (to satisfy the security requirement FPT_RVM.1).

The processes of recovering the system call of OS when the security function stops and of deleting the system call when the kernel module of RedCastle is deleted from the OS kernel at the shut-down of OS are also performed.

SFRs to be met: FPT_RVM.1

6.1.3.2 LBAC (AC.2)

The TOE provides a MAC function based on the security attributes of the subject and object. When a user logs in or a new process is created, the TOE performs a process of binding a user and a subject before allowing any TSF-mediated actions. The security attribute assigned to the subject comprises of the user's security level and security group, whose property determines the process role status (to satisfy the security requirement FIA_USB.1)

The rules of giving and inheriting a security attribute in the TOE can be divided as below according to the system status.

a) At the generation of a new process, if the upper process has a security attribute, the lower process inherits it.

b) After a user is identified and authenticated at login, the SO assigns the user the security attribute that is to be given to a user who owns the subject process. For example, if a SO with the security level of 1 logs in, the subject process is assigned (MSO, 1, 1).

c) When the privilege moves from a user to system root using su, only the role status of subject process is changed. For example, a SO with the security attribute (MSO, 1, 1) moves to system root using su, it will be changed to (SO, 1, 1).

d) A process automatically executed by init process or system root process at the system boot is assigned the security attribute based on that of the file, the object.

To give a security attribute based on the rules above, the TOE applies the following rules (to satisfy the security requirement FDP_IFC.1):

a) A process whose PID is one of 0, 1, 2, or 3 is one associated with the initial operation of system, so is assigned (SO, 1, 1).

b) If the process of system root (UXR) or security officer (SO) - PID 0, 1, 2, 3 - generates a process, the security attribute of the object is inherited to that of the subject. This intends to make sure the results performed by the system is same as those by the owner, as the daemon automatically executed at the system boot is performed by the system on behalf of its owner.

c) Considering the point of time when the uid of the subject of process is changed from 0 to the user uid as the login, give the security attribute that is assigned to a user. Give the security attribute (UXR, 0, 0) when system root logs in.

d) The security attributes of MU and UX cannot move to system root, as they are inherited without change and not related to the system management.

e) Among the security attributes of SA and SO, the group and level are inherited without change; only the role status may change according to the user status given by the system - either system root or system user.

All processes are identified and given security attributes, the label-based MAC function will operate. The TOE applies B&L(Bell & La Padula) model transformed into LBAC model. For the WRITE rule, the action is only allowed when the sensitivity labels of the subject and object are the same.

a) When a user is about to access to write a file, allow it if: Subject(level, category) = Object(level, category).

b) When a user is about to access to read a file, allow it if: Subject(level, category) ≥ Object(level, category).

Operations defined for the READ and WRITE rules for the LBAC of the TOE are:

a) READ rule: read, execute operations

b) WRITE rule: write, generate, delete operations

For efficient enforcement of the LBAC policy, the TOE compares the subject security role status with the object security role, which represents the right of the security category, instead of comparing the security category of the subject and object according to the security role status of the subject (to satisfy the security requirement FDP_IFF.2).

a) A subject with the security attribute (SO, 1, 1) or (MSO, 1, 1) is allowed to access all objects. A subject that is not in the SO group cannot access the object belonging to the SO group. The MAC between the subjects and objects that both are included in the SO group should compare their security level to determine allowance.

b) The subject of the SA group, whose role status is SA and MSA, cannot access the object of the MU group.

c) The subject of the MU group, whose role status is MU, cannot access the object of the SA group.

d) If the subject of SA group, whose role status is SA and MSA, is about to access the object of SA group, compare the level of the subject and object to determine whether to allow the access.

e) If the subject of MU group, whose role status is MU, is about to access the object of MU group, compare the levels of the subject and object to determine whether to allow the access.

f) For the object to which no security level is assigned and whose role is UX, the MAC considers it to have the lowest security attribute.

g) The TOE has a list for re-establishment of an attribute after allowing operation for the execution operation; it provides a function to re-establish the security attributes, after the subject program on the list is executed, such that it can bypass the label-based MAC regardless of the security attributes of the subject. The SO shall manage the list of the subject program that will bypass the label-based MAC to operate and the security attributes to be re-established after the operation being launched.

The TOE applies the same rule to the import and export of data, such as FTP, without special function. The data export of FTP is like the READ rule of the system call, while import is like the generation operation among the WRITE rules. Therefore, the sensitivity label is specified for the data imported based on the sensitivity of subject (to satisfy the security requirement FDP_ITC.1).

SFRs to be met: FDP_IFC.1, FDP_IFF.2, FDP_ITC.1, FIA_USB.1

6.1.3.3 RBAC (AC.3)

The system call allowed by the MAC function is transferred to the RBAC function, which enforces access control rules based on the subject's role and object's access permission.

To apply the RBAC to the object to be controlled, the administrator forms an access control policy in ACL type, which comprises the following (to satisfy the security requirement FDP_ACC.1(2)):

- Object name
- Subject information
- Operation to be allowed

The subject information necessary for the access control policy comprises:

- User role
- User identity

- User group identity
- Security group
- Subject program executed by a user
- User IP address

Operations necessary for the access between a subject and object are:

- read
- write
- execute
- create
- delete
- rename
- chmod
- chown

The user role means the role name to which the user is assigned, which the role-based policy can assign only one. The TOE enforces the following rules to determine if an operation among controlled subjects and controlled objects is allowed (to satisfy the security requirement FDP_ACF.1(2)):

a) The subject information of each operation shall be consistent with that specified in the attribute of access control of the object.

b) If a program is named, the operation can be allowed only through the program.

c) Access is allowed for the selected operation among control operations.

d) Operations other than the control operation are not enforced.

SFRs to be met: FDP_ACC.1(2), FDP_ACF.1(2)

6.1.3.4 DAC (AC.4)

The DAC based on allow/deny list provides additional control on the operations that are not included in the ACL or that require special management. The TOE provides DAC based on allow/deny list as below (to satisfy the security requirement FDP_ACC.1(1)):

a) Command control list: The command on this list will only be allowed for the SO to execute.

b) Kill prevention process list: The process on this list will only be allowed for the SO to kill.

c) Setuid execution permission list: Only those programs on this list will be allowed while the others will be denied.

d) Permitted su user list: This list specifies the user names that can change status from root to user through su. The SO can search, add to, or delete from it. The users that are not on the su control list will be denied to move from root to su.

The allow/deny list is made based on operations, which can categorize into the following two basic policies (to satisfy the security requirement FDP_ACF.1(1)):

a) Allow list based policy: The list specifies actions trusted by the system for each operation. Deny an action if it's not on the list.

b) Deny list based policy: The list specifies actions that only the SO can perform for each operation. Deny an action on the list if a user other than the SO performs it.

A system call allowed by DAC function is returned to the reference monitor and the basic system call is invoked for the function of OS to operate.

SFRs to be met: FDP_ACC.1(1), FDP_ACF.1(1)

6.1.4 Security management (SM)

6.1.4.1 Security function initiation and stop (SM.1)

The TOE provides a function to initiate or stop actions in the OS boot or shutdown, or when exceptional duties such as system maintenance are required for the following security function processing:

- Start-up and stop of RedCastle security module
- Operation modes for RedCastle security module: Warning, Enable

The SO can command initiation and stop of a security function through the GUI provided by RedCastle Manager, command interface by RedCastle Agent, and the GUI of Java interface. For initiation and stop of a security function through the GUI and Java interface, communication function of RedCastle Agent should be operating. It should be still operating when the security function stops.

At the boot and shutdown of the OS, the initiation and stop of a security function will be performed by the script of each. For automatic initiation of a security function at the boot of OS, operate the communication daemon of Agent first to perform the initiation procedures. At the shutdown of OS, stop the security function first and then terminate the daemon. The script provides a process of loading a security module into the OS before the function of the module will be initiated and a process of unloading it from the OS after the function is stopped.

While the OS is acting, the SO can require the security function of the security module to be initiated or stopped through the GUI of Manager and Java Manager and set the mode of operation of the security module as:

- Enable: MAC, RBAC, and DAC function of a security module are all operating in the case of access control violation to block access to the object and create a violation log.
- Warning: MAC, RBAC, and DAC function of a security module are all operating in the case of access control violation and create a violation log, but allow access to the object.

SFRs to be met: FMT_MOF.1, FMT_SMF.1

6.1.4.2 Security group management (SM.2)

Security group (i.e. security category) is the non-hierarchical attribute of the sensitivity label of a subject or object, which can be determined by the SO according to the property of organization. The security group in the TOE refers to the category of subject or object provided by the MAC policy, generally reflects the system area or department of the organization. The SO can manage the security group through the GUI provided by RedCastle Manager and command interface by RedCastle Agent.

- Refer to security group
- Add security group
- Delete security group
- Rename security group

The role of security group, representative of the right of security group, will be assigned at the addition of a group as below:

- Security officer (SO): Default ID is 1(Security Admin); SO group cannot be added or deleted.
- System administrator (SA): Default ID is 2(System Admins); New SA group must always be under an existing SA group. The ID will be assigned a value between 3 – 127 successively at the addition of a new security group.
- Multi-label security user (MU): No default ID is assigned; new MU group must be under an existing SO or MU group; The ID will be assigned a value between 3 – 127 successively at the addition of a new security group.

SFRs to be met: FMT_MSA.1(3), FMT_MSA.3(3), FMT_SMF.1

6.1.4.3 Security user management (SM.3)

The attributes of security user in the TOE comprise:

- User identification: user name, group name

- Security group, security level
- User role establishment
- Access control directory establishment
- Additional options: RTU(Root To User), UTR(User To Root) establishment

This function can give sensitivity label to a user and refer to, modify, and dele the security attribute of a security user. The TOE categorizes its users in terms of the subject of access control as below:

- Security user: Security level is not 0; categorized into the SO, SA, or MU according to the assigned security group.
- System user: Both security level and security group are 0.
- system root: User with uid 0, 'super user' in the existing OS; both security level and security group are 0.

The TOE categorizes its users in terms of security management as below:

- Security Manager (SO, Top Secret): A user assigned Top Secret level of security among those belong to SO group; who can perform security management and access the Management mode of Agent.
- General Manager (SO, not Top Secret): A user belonging to SO group that has the security level of other than Top Secret; who cannot perform security management and is able to access the Monitoring mode of Agent.
- User: All users that don't belong to SO group; who has no right to perform security management or search.

Security level of the security attributes is categorized into the following 6 scales:

- Top Secret (1)
- Secret (2)
- Confidential (3)
- Restricted (4)
- Unclassified (5-7)
- None (0)

The Security Manager manages the security attribute of the security user through the GUI provided by RedCastle Manager and CLI by RedCastle Agent.

SFRs to be met: FIA_ATD.1, FMT_MSA.1(3), FMT_MSA.3(3), FMT_MTD.1(2), FMT_REV.1(1), FMT_SMF.1

6.1.4.4 User role management (SM.4)

This function, to manage the roles for the enforcement of RBAC policy, provides a function to refer to, create, and delete the user role. The SO can manage the user role through the GUI provided by RedCastle Manager and CLI by RedCastle Agent.

- Refer to user role
- Add user role
- Delete user role
- Assign subject information to user role
- Delete subject information assigned to user role

Different user roles do not hold hierarchical structure. One role can be assigned 8 user IDs, 8 group IDs, 1 security group, and 1 security level, all at most. Only the security group that is assigned to is meaningful as its sub-categories are not considered.

SFRs to be met: FMT_MSA.1(2), FMT_MSA.3(2), FMT_SMF.1, FMT_SMR.2

6.1.4.5 Object security attribute management (SM.5)

The security attribute of the file, the object in the TOE, comprises the following:

- File name
- Sensitivity label
  - Security group
  - Security level

- Security role: selected by the security group

This function can give a file the sensitivity label and add, modify, and delete its security attribute. The SO manages the security attribute of the object through the GUI provided by RedCastle Manager and CLI by RedCastle Agent.

SFRs to be met: FMT_MSA.1(3), FMT_MSA.3(3), FMT_REV.1(2), FMT_SMF.1

6.1.4.6 Subject security attribute management (SM.6)

The security attribute of the process, the subject in the TOE, comprises the following:

- Process id
- Owner identity
- Sensitivity label
  - Security group
  - Security level
  - Security role status: selected by the security group and owner identity

The TOE is designed such that the security attribute of subject is automatically assigned when a new process is generated. The SO uses this function to refer to the attribute of process, the subject.

The SO refers to the security attribute of the subject through the GUI provided by RedCastle Manager and CLI by RedCastle Agent. All users except for the SO can refer to the security attribute of their processes only through the CLI.

SFRs to be met: FMT_MSA.1(3), FMT_SMF.1

6.1.4.7 Role-based policy management (SM.7)

The SO in the TOE may, for the management of RBAC policy, establish RBAC rule for each file or file group as below:

a) Subject information: Default Any
   - User
   - System group
   - Security group
   - User role

b) Subject program name: Default NULL

c) Subject IP address: Default NULL

d) Operation: Default allow access
   - read
   - write
   - execute
   - create
   - delete
   - rename
   - chmod
   - chown

This function can add or modify access control rules for one or multiple files. An access control rule may choose a subject that accesses an object and the operation the subject performs on the object. For the subject information, one item of the owner of subject, subject security group, and subject user role may be chosen; if none is chosen, it means Any (all subjects). The subject program name is the same as the subject information; if none is chosen, it means Any (all programs). The operation of a subject on an object is allowed as it is chosen as default; access is denied for operations not chosen.

The SO can manage the RBAC policy through the GUI provided by RedCastle Manager and CLI by RedCastle Agent and refer to the policy through the GUI of Java Manager provided by Agent.

- Policy group management: To add, delete, rename

- RBAC policy management: To refer to, add, delete, rename

SFRs to be met: FMT_MSA.1(2), FMT_MSA.3(2), FMT_SMF.1

6.1.4.8 Allow/deny list management (SM.8)

For the management of a policy that explicitly allow or deny subject's access to object based on the security attribute, the SO can, through the GUI provided by RedCastle Manager and CLI by RedCastle Agent, add to, refer to, and delete from the following lists:

- Command control list: specifies those commands (file names without path) that only the users of SO group can execute; the SO can refer to the list, and add or delete a new command.
- Kill prevention process list: specifies those processes (command row of operating process) that only the SO can kill; the SO can refer to, add, and delete this articles; the SO can give options to allow the system administrator to kill certain process.
- Setuid execution permission list: specifies those setuid programs (file name with path) that all users can execute; the SO can refer to, add, and delete the articles. Programs that are not on this list will be stopped execution.
- Permitted su user list: specifies the user names that can change from root to user through su; the SO can refer to, add to, or delete from the list. Users that are not on this list will be denied change from root to su.
- Execution permission command list: specifies the programs that will bypass the label-based MAC regardless of the security attribute of subject and the security attributes that will be re-established. When a program on the list executes, it will be allowed regardless of the security attribute of subject by bypassing the label-based MAC and the security attribute specified on the list will be given to the subject program allowed.

SFRs to be met: FMT_MSA.1(1), FMT_MSA.3(1), FMT_SMF.1

6.1.4.9 Audit data configuration (SM.9)

The TOE provides the SO with a function to set the place, size, and alarm of the audit storage. The SO can establish the following articles through the GUI provided by RedCastle Manager:

- Audit storage file location
- Audit storage size: File size and number
- Backup of audit storage
- Action to be taken in the case of full audit storage
- Threshold of potential violation analysis: Time interval, accumulated number of violation
- Action to be taken in the case of potential violation analysis and detection
- Email address of the administrator to be informed

SFRs to be met: FMT_MTD.1(1), FMT_SMF.1

6.1.4.10 Manager user management (SM.10)

The Manager administrator can add or delete Manager administrator and user, or modify its own Manager password through the GUI provided by RedCastle Manager.

The user first registered on RedCastle Manager at the initial execution after installation is given the right of Manager administrator, which can subsequently register and delete Manager user. Information needed for the registration of Manager user includes:

- Manager user ID
- Manager user password

The TOE provides the following functions to manage the Manager user:

a) Initial registration of Manager user (Registration of Manager administrator)

b) Registration of Manager user

c) Deletion of Manager user

d) Password change

SFRs to be met: FIA_SOS.1, FMT_MTD.1(2), FMT_MTD.1(3), FMT_REV.1(1), FMT_SMF.1

6.1.4.11 Security password management (SM.11)

The TOE provides a function to register and change the authentication data (i.e. security password) of the SO. Registration and change of security password may be done only by the user in person. The SO can change its own security password through the GUI provided by RedCastle Manager, CLI by RedCastle Agent, and GUI of Java Manager.

Authentication data is generated by combination of ID and password and encrypted using SEED and SHA-1 algorithm. To meet the SOF-Basic claim, the password selected at the registration and modification of security password must satisfy the following conditions:

- Password length: 8 – 15 byte
- Acceptable password characters are:
  - 52 alphabetic letters (lowercase or uppercase)
  - 10 numeric digits (0-9)
  - 16 special characters (!, @, #, $, %, ^, &, *, (,), +, =, <, >, :, ;)
- Password must contain at least one alphabetic letter, numeric digit, and special character.
- Consecutive numeric or alphabetic characters are allowed.
- Repeated use of same alphabetic, numeric, and special characters are allowed.

SFRs to be met: FIA_SOS.1, FMT_MTD.1(3), FMT_SMF.1

6.1.4.12 Security function configuration (SM.12)

After the security function is initiated, the TOE provides the SO, through the GUI, CLI, and JAVA GUI, with a function to configure for the operation of security function as below:

- LBAC function: On, Warning
  - Options for inheriting security attribute at the generation of an object: On, Off
- RBAC function: On, Warning, Off
- Operation mode of command control: On, Warning, Off
- Operation mode of SETUID execution permission: On, Warning, Off
- Operation mode of Kill control: On, Warning, Off
- RTU options: On, Off
- UTR options: On, Warning, Off

SFRs to be met: FMT_MTD.1(4), FMT_SMF.1

6.1.5 TSF protection (TP)

6.1.5.1 Abstract machine and TSF operation testing (TP.1)

The TOE provides an abstract machine and TSF operation testing function to check the state of OS of the system on which RedCastle Agent is installed and operating and the operation of the TOE.

Abstract machine and TSF operation testing is performed periodically through the GUI provided by RedCastle Manager and includes the following functions:

- System state review: Consumed CPU, consumed memory, elapsed boot time
- Agent state review: Manager login period, security module version, operation state of the security module, log daemon, and iptables
- Real-time security log review

- License: The TOE provides function separately in Basic mode and Advanced mode according to the license. Basic mode starts right after the TOE is installed; only with license Advanced mode starts and all functions are activated.
- System information review: Basic system information, partition information, hardware information, software information (which is a non-security function)

SFRs to be met: FMT_SMF.1, FPT_AMT.1, FPT_TST.1

6.1.5.2 Integrity check and management (TP.2)

For secure operation of the TSF, the TOE provides a function to check the integrity of the TSF execution file, TSF data file, and file that the administrator specifies at the start-up, periodically during normal operation, and when the SO requires.

The SO can perform the following functions through the GUI provided by RedCastle Manager:

- Register file that will be checked for integrity
- Delete file from the integrity check list
- Perform integrity check
- Review the results of integrity check and update
- Set the interval of integrity check (only provided in the Agent integrity check function)

The execution files of RedCastle Manager and RedCastle Agent, which consist of the TOE, will be registered on the list for integrity check and cannot be deleted from it.

The integrity check value will be stored at the first check, and a new value obtained from the next check will be compared with that stored before. The TOE uses SHA-1 for the integrity check.

SFRs to be met: FMT_MTD.1(4), FMT_SMF.1, FPT_TST.1

6.1.6 TOE access (TA)

6.1.6.1 Service control (TA.1)

In addition to the identification and authentication of the OS, the TOE provides a separate control mechanism for the terminal services provided by Asianux Server 3, such as dtlogin, ftp, rlogin, ssh, telnet, rsh, and su. The TOE provides service control function based on PAM provided by Asianux Server 3.

The SO manages the service and session control policy by user and group and the service and session control policy for all users through the GUI provided by RedCastle Manager.

Service control policy by user and group comprises the following:

- Allow/deny service control policy
- Services to be controlled: All, rexec, rsh, rlogin, su, ftp, remote. sshd
- Access IP address
- Access time scope: hour, day, week, month

Session control policy by user and group comprises the following:

- Allow all sessions
- The allowed number of session: 0 ~ 99
- Console login will always be allowed

The service and session control policy for all users includes the policy for each user and group, and adds a policy to limit the number of session for a specific IP address.

- IP address for which the number of session will be limited
- Allowed number of session: -1 ~ 9999 (except that -1 will not be allowed a session)

The TOE provides GUI, as one of the non-security functions, with which the authorized can perform management function of system user and group provided by Asianux Server 3 through RedCastle Manager.

- Add/modify/delete system account information

- Establish the policy of system account password
- Modify system account password
- Add/modify/delete system group

SFRs to be met: FMT_SMF.1, FPT_LSA.1, FTA_MCS.1, FTA_TSE.1

6.1.6.2 Manager screen saver (TA.2)

The TSF can lock an interacting session after the SO's inactivity and unlock the session using identification and authentication of RedCastle Manager. The TOE provides session locking during inactivity through the screen saver of Manager.

This function applies to the Windows system on which RedCastle Manager is operating and allows setting the waiting time using GUI. Screen saver works when the time of user's inactivity exceeds the waiting time. Screen will be unlocked only if the Manager user enters the login password.

SFRs to be met: FMT_MTD.1(4), FMT_SMF.1, FTA_SSL.1

6.2 Assurance measures

[Table 6-1] Mapping assurance measures to assurance components

Assurance component	How the assurance is provided	Assurance measures
ACM_AUT.1	The TOE uses CVS as an automated tool; the Configuration Management describes the CM plan and automated tools.	RedCastle v3.0 for Asianux Server 3 Configuration Management V1.3 (“Configuration Management”)
ACM_CAP.4	The implementation representation of the TOE is managed using the CM system; the Configuration Management identifies the CM list and describes the CM plan and acceptance plan.	
ACM_SCP.2	The Configuration Management identifies the CM list for security flaws and provides description of problem tracking tools and evidence.	
ADO_DEL.2	Describes the TOE delivery procedures and all those procedures necessary to maintain security when distributing the TOE.	RedCastle v3.0 for Asianux Server 3 Delivery Procedures V1.4 (“Delivery Procedures”)
ADO_IGS.1	Describes the procedures necessary for secure installation, generation, and start-up of the TOE.	RedCastle v3.0 for Asianux Server 3 Installation Guide V1.2 (“Installation Guide”)
ADV_FSP.2	The Functional Specification describes the TSF and its external interfaces using an informal style; identifies all external TSF interfaces, providing complete details.	RedCastle v3.0 for Asianux Server 3 Functional Specification V1.3 (“Functional Specification”)
ADV_HLD.2	The High-level Design separates security function and non-security function; describes the structure of the TSF in terms of subsystems and the security functionality provided by each subsystem in an informal	RedCastle v3.0 for Asianux Server 3 High-level Design V1.4 (“High-level Design”)

	style.	
ADV_IMP.1	Unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.	RedCastle v3.0 for Asianux Server 3 Implementation Validation V1.3 (“Implementation Validation”)
ADV_LLD.1	Describes the TSF in terms of modules; describes the purpose of each module, how each function is provided, and relationship between modules; identifies all interfaces to the modules and which are externally visible interfaces to the modules.	RedCastle v3.0 for Asianux Server 3 Low-level Design V1.3 (“Low-level Design”)
ADV_RCR.1	Analyzes correspondence between the SFRs and SFs; between the SFs and subsystems; between the subsystems and modules; between the modules and implementation representation.	Functional Specification, High-level Design, Low-level Design, and Implementation Validation providing correspondence analysis
ADV_SPM.1	Provides the TSP model and demonstrates correspondence between the functional specification and TSP model.	RedCastle v3.0 for Asianux Server 3 Security Policy Modeling V1.3 (“Security Policy Modeling”)
AGD_ADM.1	Provides administrator guidance addressed to system administrative personnel; describes the functions and interfaces available to the administrator of the TOE.	RedCastle v3.0 for Asianux Server 3 Administration Manual V1.2 (“Administration Manual”)
AGD_USR.1	This component is not provided as there is no functions or interfaces in the TOE available to the non-administrative users of the TOE.	(Not provided)
ALC_DVS.1	Describes the procedures and security measures necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.	RedCastle v3.0 for Asianux Server 3 Life Cycle Support V1.3 (“Life Cycle Support”)

ALC_LCD.1	Establishes a life-cycle model to be used in the development and maintenance of the TOE and provides life-cycle definition documentation.	
ALC_TAT.1	Identifies the development tools being used for the TOE.	
ATE_COV.2	Analyzes the test coverage and demonstrates the correspondence between the tests identified and the TSF described in the Functional Specification.	RedCastle v3.0 for Asianux Server 3 Tests V1.2 (“Test documents”)
ATE_DPT.1	Provides an analysis that the tests identified are sufficient to demonstrate that the TSF operates in accordance with its high-level design.	
ATE_FUN.1	Tests the TSF and document the results; provides test documentation consisting of test plans, test procedure descriptions, expected test results, and actual test results.	
ATE_IND.2	Provides the evaluator with the TOE for testing.	N/A (Evaluator)
AVA_MSU.2	The Guidance Documentation identifies modes of operation of the TOE, their consequences and implications for maintaining secure operation; lists all requirements for external security measures.	RedCastle v3.0 for Asianux Server 3 Misuse Analysis V1.2 (“Misuse analysis”)
AVA_SOF.1	Performs a strength of TOE security function analysis for each mechanism with a strength of TOE security function claim to show that it meets or exceeds the strength of function metric.	RedCastle v3.0 for Asianux Server 3 Vulnerability Analysis V1.2 (“Vulnerability analysis”)
AVA_VLA.2	Identifies vulnerabilities; performs an analysis that shows the TOE is resistant to the identified vulnerabilities and documents the results.	

7 PP claims

This ST does not claim conformance with any PP; the security functions and assurance measures of the TOE do not conform to a PP.

8 Rationale

This chapter describes the security objectives that are defined based on the security environment (threats, assumptions, OSPs), the security requirements that meet the security objectives, and the TOE summary specification rationale that shows the the TOE security functions meet the security requirements. The rationale shows that the TOE provides an effective set of IT security countermeasures within the security environment.

8.1 Security objectives rationale

Security objectives rationale shows that the specified security objectives are suitable, sufficient to address security problem, and not onerous.

Security objectives rational demonstrates:

- that each assumption, threat, and OSP is addressed by at least one security objective.
- that each security objective addresses at least one assumption, threat, and OSP.

[Table 8-1] Mapping security objectives to the security environment

Security objective / Security requirement	Security objectives for the TOE									Security objectives for the environment					
	O. Audit	O.Management	O.IA	O.Data	O.Securitylevel	O.Securityrole	O.DAC	O.MAC	O.RBAC	OE.Locate	OE.Administrator	OE.Securemanagement	OE.OSpatch	OE.SSLprotocol	OE.Timestamp
A.Locate										○					
A.Administrator											○				
A.OSpatch													○		
A.Installation												○			
A.SSLprotocol														○	

A.Timestamp															○
T.Data			○	○											
T.Masquerade			○												
T.Bypass							○	○	○						
T.Integrity														○	
TE.Delivery												○			
P.Audit	○														
P.IA			○												
P.Management		○										○			
P.Securitylevel					○										
P.Securityrole						○									
P.DAC							○								
P.RBAC									○						
P.MAC								○							

8.1.1 Rationale for the security objectives for the TOE

O.Audit

This is necessary to support P.Audit because it ensures that the TOE provides a means of recording and reviewing security-relevant events.

O.Management

This is necessary to support P.Management because it ensures that the TOE provides the administrator with a means to access and manage the TOE securely.

O.IA

The identification and authentication function of the TOS must be able to prevent an unauthorized user from accessing the TOE. This objective is necessary to counter T.Data and T.Masquerade and to support P.IA because it ensures that a user must be identified and authenticated to access the TOE or information.

O.Data

This is necessary to counter T.Data because it ensures that the TOE protects the TOE data or reliable data from unauthorized tampering.

O.Securitylevel

This satisfies P.Securitylevel because it ensures that the TOE assigns and annuls an appropriate security level of a subject and object according to the access control policy and procedures of the organization.

O.Securityrole

This is necessary to support P.Securityrole because it ensures that the TOE provides a means the authorized administrator to generate, delete, manage, and review roles in accordance with RBAC policy, which reduces threat that can be encountered due to incorrect assignment and management of user's roles.

O.DAC

This is necessary to counter T.Bypass and support P.DAC because it ensures that the TOE cannot be bypassed when a user accesses the resources and that the TOE controls access to the resources based on the user identity.

O.MAC

This is necessary to counter T.Bypass and supports P.MAC because it ensures that the TOE cannot be bypassed when a user accesses the resources and that the TOE controls access to information based on the user identity.

O.RBAC

This is necessary to counter T.Masquerade and T.Bypass and to support P.RBAC because it ensures that the TOE cannot be bypassed when a user accesses the resources and that the TOE controls access to object based on the user roles.

8.1.2 Rationale for the security objectives for the environment

OE.Locate

This addresses A.Locate because it ensures that the TOE is located in a physically secure environment where only an authorized administrator can access.

OE.Administrator

This addresses A.Administrator because it ensures that the administrator of the TOE is reliable and able to administer the TOE securely.

OE.Securemanagement

This counters TE.Delivery and satisfies A.Installation and P.Management because it ensures that the TOE is delivered and installed securely, and configured, managed, and used in a secure manner by an authorized administrator.

OE.OSpatch

This addresses A.OSpatch because it ensures that the TOE deletes useless services or means from the OS where it is installed and patches the OS on its vulnerabilities so that the confidence and reliability of the OS can be secured.

OE.SSLprotocol SSL

This addresses A.SSLprotocol and counters T.Integrity because it ensures that the TOE uses SSL protocol to provide secure communication between RedCastle Manager and RedCastle Agent.

OE.Timestamp

This addresses A.Timestamp because it ensures the reliability of the timestamp that is provided for the TSF in the commodity OS.

8.2 Security requirements rationale

This chapter demonstrates that the described IT security requirements are suitable to meet the security objectives and, consequently, to address security problem.

8.2.1 Rational for the TOE security functional requirements

This demonstrates that:

- Each security objective for the TOE is addressed by at least one TOE SFR.
- Each TOE SFR addresses at least one security objective for the TOE.

[Table 8-2] Mapping SFRs to the security objectives

Security Objective / SFR	Security objectives for the TOE								
	O.Audit	O.Management	O.IA	O.Data	O.Securitylevel	O.Securityrole	O.DAC	O.MAC	O.RBAC
FAU_ARP.1	○								
FAU_GEN.1	○								
FAU_GEN.2	○								
FAU_SAA.1	○								
FAU_SAR.1	○								
FAU_SAR.2	○								
FAU_SAR.3	○								
FAU_SEL.1	○								
FAU_STG.1	○								
FAU_STG.3	○								
FAU_STG.4	○								
FDP_ACC.1(1)							○		
FDP_ACC.1(2)									○
FDP_ACF.1(1)							○		
FDP_ACF.1(2)									○

FDP_IFC.1								○	
FDP_IFF.2								○	
FDP_ITC.1								○	
FIA_AFL.1			○						
FIA_ATD.1			○						
FIA_SOS.1			○						
FIA_UAU.2		○	○	○					
FIA_UAU.7			○						
FIA_UID.2		○	○	○					
FIA_USB.1					○				
FMT_MOF.1		○							
FMT_MSA.1(1)		○					○		
FMT_MSA.1(2)		○				○			○
FMT_MSA.1(3)		○			○			○	
FMT_MSA.3(1)		○					○		
FMT_MSA.3(2)		○				○			○
FMT_MSA.3(3)		○			○			○	
FMT_MTD.1(1)		○							
FMT_MTD.1(2)		○							
FMT_MTD.1(3)		○							
FMT_MTD.1(4)		○							
FMT_REV.1(1)		○				○			
FMT_REV.1(2)		○			○				
FMT_SMF.1		○							
FMT_SMR.2		○				○			○
FPT_AMT.1		○							
FPT_RVM.1							○	○	○
FPT_TST.1				○					
FTA_LSA.1		○	○						
FTA_MCS.1		○	○						
FTA_SSL.1			○						
FTA_TSE.1		○	○						

FAU_ARP.1 Security alarms

This component satisfies **O.Audit** because it ensures a function to take actions upon detection of a potential security violation.

FAU_GEN.1 Audit data generation

This component satisfies **O.Audit** because it ensures a function to define auditable events and generate audit records.

FAU_GEN.2 User identity association

This component satisfies **O.Audit** because it ensures a function to associate each auditable event with the identity of the user that caused the event.

FAU_SAA.1 Potential violation analysis

This component satisfies **O.Audit** because it ensures a function to monitor the audited events and indicate a potential violation of the TSP.

FAU_SAR.1 Audit review

This component satisfies **O.Audit** because it ensures a function to provide the authorized administrator with the capability to read audit data from the audit records.

FAU_SAR.2 Restricted audit review

This component satisfies **O.Audit** because it ensures a function to prohibit all users read access to the audit records, except the authorized administrator that has been granted explicit read-access.

FAU_SAR.3 Selectable audit review

This component satisfies **O.Audit** because it ensures the ability to perform searches and sorting of audit data based on criteria with logical relations.

FAU_SEL.1 Selective audit

This component satisfies **O.Audit** because it ensures a function to include or exclude auditable events from the set of audited events based on the defined attributes.

FAU_STG.1 Protected audit trail storage

This component satisfies **O.Audit** because it ensures a function to protect the stored audit records from unauthorized deletion and modification.

FAU_STG.3 Action in case of possible audit data loss

This component satisfies **O.Audit** because it ensures a function to take actions if the audit trail exceeds predefined limit.

FAU_STG.4 Prevention of audit data loss

This component satisfies **O.Audit** because it ensures a function to take actions if the audit trail is full.

FDP_ACC.1(1) Subset access control

This component satisfies **O.DAC** because it ensures a function to enforce the DAC policy and to define the scope to be covered by the policy.

FDP_ACC.1(2) Subset access control

This component satisfies **O.RBAC** because it ensures a function to enforce the RBAC policy and to define the scope to be covered by the policy.

FDP_ACF.1(1) Security attribute based access control

This component satisfies **O.DAC** because it ensures a function to enforce the DAC policy based on the security attributes.

FDP_ACF.1(2) Security attribute based access control

This component satisfies **O.RBAC** because it ensures a function to enforce the RBAC policy based on the security attributes.

FDP_IFC.1 Subset information flow control

This component satisfies **O.MAC** because it ensures a function to enforce the MAC policy and to define the scope to be covered by the policy.

FDP_IFF.2 Hierarchical security attributes

This component satisfies **O.MAC** because it ensures a function to enforce the MAC policy based on the security attributes.

FDP_ITC.1 Import of user data without security attributes

This component satisfies **O.MAC** because it ensures a function to enforce the MAC and DAC policy when importing user data without sensitivity label.

FIA_AFL.1 Authentication failure handling

This component satisfies **O.IA** because it ensures a function to define the number of failed authentication attempts of the authorized administrator and to take actions when the defined number is met or surpassed.

FIA_ATD.1 User attribute definition

This component satisfies **O.IA** because it ensures a function to maintain the list of security attributes belonging to individual users.

FIA_SOS.1 Verification of secrets

This component satisfies **O.IA** because it ensures a function to provide a mechanism to verify that secrets meet the defined quality metric.

FIA_UAU.2 User authentication before any action

This component satisfies **O.Management**, **O.Data**, and **O.IA** because it ensures a function to successfully authenticate each user before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.7 Protected authentication feedback

This component satisfies **O.IA** because it ensures a function to provide only the specified list of feedback to the user while the authentication is in progress.

FIA_UID.2 User identification before any action

This component satisfies **O.Management**, **O.Data**, and **O.IA** because it ensures a function to identify each user before allowing any other TSF-mediated actions on behalf of that user.

FIA_USB.1 User-subject binding

This component satisfies **O.Securitylevel** because it ensures a function to associate the user security attributes with subjects acting on behalf of that user.

FMT_MOF.1 Management of security functions behavior

This component satisfies **O.Management** because it ensures the ability of the authorized administrator to manage the security functions.

FMT_MSA.1(1) Management of security attributes

This component satisfies **O.Management** and **O.DAC** because it ensures the ability of the authorized administrator to manage the security attributes to be covered by the DAC policy.

FMT_MSA.1(2) Management of security attributes

This component satisfies **O.Management**, **O.Securityrole**, and **O.RBAC** because it ensures the ability of the authorized administrator to manage the security attributes to be covered by the RBAC policy.

FMT_MSA.1(3) Management of security attributes

This component satisfies **O.MAC**, **O.Management**, and **O.Securitylevel** because it ensures the ability of the authorized administrator to manage the security attributes to be covered by the DAC policy.

FMT_MSA.3(1) Static attribute initialization

This component satisfies **O.Management** and **O.DAC** because it ensures a function to provide default values for security attributes that are used to enforce the DAC policy.

FMT_MSA.3(2) Static attribute initialization

This component satisfies **O.Management**, **O.Securityrole**, and **O.RBAC** because it ensures a function to provide default values for security attributes that are used to enforce the RBAC policy.

FMT_MSA.3(3) Static attribute initialization

This component satisfies **O.MAC**, **O.Management**, and **O.Securitylevel** because it ensures a function to provide default values for security attributes that are used to enforce the MAC policy.

FMT_MTD.1(1) Management of TSF data

This component satisfies **O.Management** because it ensures restricting the ability to manage the audit data to the authorized administrator.

FMT_MTD.1(2) Management of TSF data

This component satisfies **O.Management** because it ensures restricting the ability to delete and initialize the identification and authentication data to the authorized administrator.

FMT_MTD.1(3) Management of TSF data

This component satisfies **O.Management** because it ensures restricting the ability to modify the authentication data to the authorized administrator and the user authorized to modify its own authentication data.

FMT_MTD.1(4) Management of TSF data

This component satisfies **O.Management** because it ensures restricting the ability to manage the TSF data associated with security to the authorized administrator.

FMT_REV.1(1) Revocation

This component satisfies **O.Management** and **O.Securityrole** because it ensures a function to restrict the ability to revoke security attributes associated with the users to the authorized administrator.

FMT_REV.1(2) Revocation

This component satisfies **O.Management** and **O.Securitylevel** because it ensures a function to restrict the ability to revoke security attributes associated with the objects to the authorized administrator.

FMT_SMF.1 Specification of management functions

This component satisfies **O.Management** because it ensures the capability of performing the security management functions of security attributes, TSF data, security functions, etc.

FMT_SMR.2 Restrictions on security roles

This component satisfies **O.Management**, **O.Securityrole**, and **O.RBAC** because it ensures that the users and user roles are managed and that the conditions for the user roles are satisfied.

FPT_AMT.1 Abstract machine testing

This component satisfies **O.Management** because it ensures a function to run a suite of tests to demonstrate the correct operation of the abstract machine that underlies the TSF.

FPT_RVM.1 Non-bypassability of the TSP

This component satisfies **O.DAC**, **O.MAC**, and **O.RBAC** because it ensures that TSP enforcement functions are invoked and succeed.

FPT_TST.1 TSF testing

This component satisfies **O.Data** because it ensures a function to run a suite of self tests to demonstrate the correct operation of the TSF and to provide the authorized administrator with the capability to verify the integrity of the TSF data and TSF executable code.

FTA_LSA.1 Limitation on scope of selectable attributes

This component satisfies **O.Management** and **O.IA** because it ensures a function to restrict the scope of the session security attributes that can be chosen for that session while a user is being identified and authenticated.

FTA_MCS.1 Basic limitation on multiple concurrent sessions

This component satisfies **O.Management** and **O.IA** because it ensures a function to restrict the maximum number of concurrent sessions that belong to the same user while the user is being identified and authenticated.

FTA_SSL.1 TSF-initiated session locking

This component satisfies **O.IA** because it ensures a function to lock an interactive session after time interval of the authorized administrator inactivity and require authentication of the authorized administrator prior to unlocking the session.

FTA_TSE.1 TOE session establishment

This component satisfies **O.Management** and **O.IA** because it ensures a function to deny a user permission to establish a session with the TOE.

8.2.2 Rationale for the TOE assurance requirements

This ST provides fully defined external interfaces specification and low-level design as the assurance measures for the analysis of the TOE security function. It also provides another assurance measures, to support the analysis, like independent testing and independent vulnerability analysis demonstrating resistance to penetration attackers with a low attack potential.

The TOE ensures EAL4 because it satisfies the requirements and provides a moderate level of independently assured security.

8.2.3 Rationale for the security requirements for the IT environment

[Table 8-3] Mapping security requirements for the IT environment to security objectives for the environment

Security objective / SFR	OE.SSLprotocol	OE.Timestamp
FPT_ITT.1	○	
FPT_STM.1		○

FPT_STM.1 Reliable time stamps

This component satisfies **OE.Timestamp**, which requires the IT environment of the TOE to provide timestamp in the commodity OS, because it ensures that the TOE provides reliable time stamps for its own use.

FPT_ITT.1 Basic internal TSF data transfer protection

This component satisfies **OE.SSLprotocol**, which requires that the TSF data be protected while being transmitted, because it protects TSF data from disclosure and modification when it is transmitted between separate parts of the TOE.

8.3 Dependencies rationale

8.3.1 Dependencies between the TOE security functional requirements

[Table 8-4] summarizes the dependencies between the TOE security functional components.

- FDP_IFC.1 has dependencies on FDP_IFF.1, which will be satisfied by FDP_IFF.2 hierarchical to FDP_IFF.1.
- FAU_GEN.2, FIA_UAU.2, and FTA_MCS.1 have dependencies on FIA_UID.1, which will be satisfied by FIA_UID.2 hierarchical to FIA_UID.2.
- FIA_AFL.1, FIA_UAU.7, and FTA_SSL.1 have dependencies on FIA_UAU.1, which will be satisfied by FIA_UAU.2 hierarchical to FIA_UAU.1.
- FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, and FMT_REV.1 have dependencies on FMT_SMR.1, which will be satisfied by FMT_SMR.2 hierarchical to FMT_SMR.1.

[Table 8-4] Dependencies between functional components

No.	Functional component	Dependencies	Reference
1	FAU_ARP.1	FAU_SAA.1	4
2	FAU_GEN.1	FPT_STM.1	50
3	FAU_GEN.2	FAU_GEN.1	2

		FIA_UID.1	25
4	FAU_SAA.1	FAU_GEN.1	2
5	FAU_SAR.1	FAU_GEN.1	2
6	FAU_SAR.2	FAU_SAR.1	5
7	FAU_SAR.3	FAU_SAR.1	5
8	FAU_SEL.1	FAU_GEN.1 FMT_MTD.1	2 34, 35, 36, 37
9	FAU_STG.1	FAU_GEN.1	2
10	FAU_STG.3	FAU_STG.1	9
11	FAU_STG.4	FAU_STG.1	9
12	FDP_ACC.1(1)	FDP_ACF.1	14
13	FDP_ACC.1(2)	FDP_ACF.1	15
14	FDP_ACF.1(1)	FDP_ACC.1 FMT_MSA.3	12 31
15	FDP_ACF.1(2)	FDP_ACC.1 FMT_MSA.3	13 32
16	FDP_IFC.1	FDP_IFF.1	17
17	FDP_IFF.2	FDP_IFC.1 FMT_MSA.3	16 33
18	FDP_ITC.1	[FDP_IFC.1] FMT_MSA.3	16 33
19	FIA_AFL.1	FIA_UAU.1	22
20	FIA_ATD.1	-	-
21	FIA_SOS.1	-	-
22	FIA_UAU.2	FIA_UID.1	25
23	FIA_UAU.7	FIA_UAU.1	22
24	FIA_UID.2	-	-
25	FIA_USB.1	FIA_ATD.1	20
26	FMT_MOF.1	FMT_SMR.1 FMT_SMF.1	40 39
27	FMT_MSA.1(1)	[FDP_ACC.1] FMT_SMR.1 FMT_SMF.1	12 40 39
28	FMT_MSA.1(2)	[FDP_ACC.1]	13

		FMT_SMR.1	40
		FMT_SMF.1	39
29	FMT_MSA.1(3)	[FDP_IFC.1]	16
		FMT_SMR.1	40
		FMT_SMF.1	39
30	FMT_MSA.3(1)	FMT_MSA.1	28
		FMT_SMR.1	40
31	FMT_MSA.3(2)	FMT_MSA.1	29
		FMT_SMR.1	40
32	FMT_MSA.3(3)	FMT_MSA.1	30
		FMT_SMR.1	40
33	FMT_MTD.1(1)	FMT_SMR.1	40
		FMT_SMF.1	39
34	FMT_MTD.1(2)	FMT_SMR.1	40
		FMT_SMF.1	39
35	FMT_MTD.1(3)	FMT_SMR.1	40
		FMT_SMF.1	39
36	FMT_MTD.1(4)	FMT_SMR.1	40
		FMT_SMF.1	39
37	FMT_REV.1(1)	FMT_SMR.1	40
38	FMT_REV.1(2)	FMT_SMR.1	40
39	FMT_SMF.1	-	-
40	FMT_SMR.2	FIA_UID.1	25
41	FPT_AMT.1	-	-
42	FPT_RVM.1	-	-
43	FPT_TST.1	FPT_AMT.1	41
44	FTA_LSA.1		
45	FTA_MCS.1	FIA_UID.1	25
46	FTA_SSL.1	FIA_UAU.1	22
47	FTA_TSE.1	-	-
48	FPT_ITT.1	-	-
49	FPT_STM.1	-	-

8.3.2 Dependencies between the TOE assurance requirements

The following table summarizes the dependencies between the TOE security assurance components.

[Table 8-5] Dependencies between assurance components

No.	Assurance component	Dependencies	Reference
1	ACM_AUT.1	ACM_CAP.3	2
2	ACM_CAP.4	ALC_DVS.1	14
3	ACM_SCP.2	ACM_CAP.3	2
4	ADO_DEL.2	ACM_CAP.3	2
5	ADO_IGS.1	AGD_ADM.1	12
6	ADV_FSP.2	ADV_RCR.1	10
7	ADV_HLD.2	ADV_FSP.1 ADV_RCR.1	6 10
8	ADV_IMP.1	ADV_LLD.1 ADV_RCR.1 ALC_TAT.1	9 10 16
9	ADV_LLD.1	ADV_HLD.2 ADV_RCR.1	7 10
10	ADV_RCR.1	-	
11	ADV_SPM.1	ADV_FSP.1	6
12	AGD_ADM.1	ADV_FSP.1	6
13	ADG_USR.1	ADV_FSP.1	6
14	ALC_DVS.1	-	
15	ALC_LCD.1	-	
16	ALC_TAT.1	ADV_IMP.1	8
17	ATE_COV.2	ADV_FSP.1 ATE_FUN.1	6 19
18	ATE_DPT.1	ADV_HLD.1 ATE_FUN.1	7 19
19	ATE_FUN.1	-	
20	ATE_IND.2	ADV_FSP.1 AGD_ADM.1	6 12

		AGD_USR.1	13
		ATE_FUN.1	19
21	AVA_MSU.2	ADO_IGS.1	5
		ADV_FSP.1	6
		AGD_ADM.1	12
		AGD_USR.1	13
22	AVA_SOF.1	ADV_FSP.1	6
		ADV_HLD.1	7
23	AVA_VLA.2	ADV_FSP.1	6
		ADV_HLD.2	7
		ADV_IMP.1	8
		ADV_LLD.1	9
		AGD_ADM.1	12
		AGD_USR.1	13

8.4 TOE summary specification rationale

8.4.1 Conformance with the TOE security functions

The following table shows the security functions specified in the TOE summary specification.

[Table 8-6] TOE security functions

ID	Security function	ID	Security function
AU.1	Audit data generation and collection	SM.5	Object security attribute management
AU.2	Potential violation analysis and action	SM.6	Subject security attribute management
AU.3	Audit storage management	SM.7	Role-based policy management
AU.4	Audit data reference and review	SM.8	Allow/deny list management
IA.1	Manager identification and authentication	SM.9	Audit data configuration
IA.2	Agent identification and authentication	SM.10	Manager user management
AC.1	Reference monitor	SM.11	Security password management
AC.2	LBAC	SM.12	Security function configuration
AC.3	RBAC	TP.1	Abstract machine and TSF operation testing
AC.4	DAC	TP.2	Integrity check and management

SM.1	Security function initiation and stop	TA.1	Service control
SM.2	Security group management	TA.2	Manager screen saver
SM.3	Security user management		
SM.4	User role management		

The following table shows the relationship between the security functions described in the TOE summary specification and the TOE security functional requirements.

[Table 8-7] Mapping TOE SFRs to the security functions in the TSS

TOE SFR			Security function in the TOE summary specification
Class	Component	Element	
Security audit	FAU_ARP.1	FAU_ARP.1.1	AU.2
	FAU_GEN.1	FAU_GEN.1.1	AU.1
		FAU_GEN.1.2	AU.1
	FAU_GEN.2	FAU_GEN.2.1	AU.1
	FAU_SAA.1	FAU_SAA.1.1	AU.2
		FAU_SAA.1.2	AU.2
	FAU_SAR.1	FAU_SAR.1.1	AU.4
		FAU_SAR.1.2	AU.4
	FAU_SAR.2	FAU_SAR.2.1	AU.3
	FAU_SAR.3	FAU_SAR.3.1	AU.4
	FAU_SEL.1	FAU_SEL.1.1	AU.1
	FAU_STG.1	FAU_STG.1.1	AU.3
		FAU_STG.1.2	AU.3
	FAU_STG.3	FAU_STG.3.1	AU.3
	FAU_STG.4	FAU_STG.4.1	AU.3
User data protection	FDP_ACC.1(1)	FDP_ACC.1.1	AC.4
	FDP_ACC.1(2)	FDP_ACC.1.1	AC.3
	FDP_ACF.1(1)	FDP_ACF.1.1	AC.4
		FDP_ACF.1.2	AC.4
		FDP_ACF.1.3	AC.4
		FDP_ACF.1.4	AC.4

	FDP_ACF.1(2)	FDP_ACF.1.1	AC.3
		FDP_ACF.1.2	AC.3
		FDP_ACF.1.3	AC.3
		FDP_ACF.1.4	AC.3
	FDP_IFC.1	FDP_IFC.1.1	AC.2
	FDP_IFF.2	FDP_IFF.2.1	AC.2
		FDP_IFF.2.2	AC.2
		FDP_IFF.2.3	AC.2
		FDP_IFF.2.4	AC.2
		FDP_IFF.2.5	AC.2
		FDP_IFF.2.6	AC.2
		FDP_IFF.2.7	AC.2
	FDP_ITC.1	FDP_ITC.1.1	AC.2
		FDP_ITC.1.2	AC.2
		FDP_ITC.1.3	AC.2
Identification and authentication	FIA_AFL.1	FIA_AFL.1.1	IA.2
		FIA_AFL.1.2	IA.2
	FIA_ATD.1	FIA_ATD.1.1	SM.3
	FIA_SOS.1	FIA_SOS.1.1	SM.10, SM.11
	FIA_UAU.2	FIA_UAU.2.1	IA.1, IA.2
	FIA_UAU.7	FIA_UAU.7.1	IA.1, IA.2
	FIA_UID.2	FIA_UID.2.1	IA.1, IA.2
	FIA_USB.1	FIA_USB.1.1	AC.2
		FIA_USB.1.2	AC.2
		FIA_USB.1.3	AC.2
Security management	FMT_MOF.1	FMT_MOF.1.1	SM.1
	FMT_MSA.1(1)	FMT_MSA.1.1	SM.8
	FMT_MSA.1(2)	FMT_MSA.1.1	SM.4, SM.7
	FMT_MSA.1(3)	FMT_MSA.1.1	SM.2, SM.3, SM.5, SM.6
	FMT_MSA.3(1)	FMT_MSA.3.1	SM.8
		FMT_MSA.3.2	SM.8
	FMT_MSA.3(2)	FMT_MSA.3.1	SM.4, SM.7
		FMT_MSA.3.2	SM.4, SM.7
	FMT_MSA.3(3)	FMT_MSA.3.1	SM.2, SM.3, SM.5
		FMT_MSA.3.2	SM.2, SM.3, SM.5

	FMT_MTD.1(1)	FMT_MTD.1.1	SM.9
	FMT_MTD.1(2)	FMT_MTD.1.1	SM.3, SM.10
	FMT_MTD.1(3)	FMT_MTD.1.1	SM.10, SM.11
	FMT_MTD.1(4)	FMT_MTD.1.1	SM.12, TP2, TA.2
	FMT_REV.1(1)	FMT_REV.1.1	SM.3, SM.10
		FMT_REV.1.2	SM.3, SM.10
	FMT_REV.1(2)	FMT_REV.1.1	SM.5
		FMT_REV.1.2	SM.5
	FMT_SMF.1	FMT_SMF.1.1	SM.1, SM.2, SM.3, SM.4, SM.5, SM.6, SM.7, SM.8, SM.9, SM.10, SM.11, SM.12, TP.1,TP.2, TA.1, TA.2
	FMT_SMR.2	FMT_SMR.2.1	SM.4
		FMT_SMR.2.2	SM.4
		FMT_SMR.2.3	SM.4
Protection of the TSF	FPT_AMT.1	FPT_AMT.1.1	TP.1
	FPT_RVM.1	FPT_RVM.1.1	AC.1
	FPT_TST.1	FPT_TST.1.1	TP.1
		FPT_TST.1.2	TP.2
		FPT_TST.1.3	TP.2
TOE access	FTA_LSA.1	FTA_LSA.1.1	TA.1
	FTA_MCS.1	FTA_MCS.1.1	TA.1
		FTA_MCS.1.2	TA.1
	FTA_SSL.1	FTA_SSL.1.1	TA.2
		FTA_SSL.1.2	TA.2
	FTA_TSE.1	FTA_TSE.1.1	TA.1

AU.1 Audit data generation and collection

This SF satisfies **FAU_GEN.1 Audit Data Generation** because it provides a function to define auditable events and generate audit records.

This SF satisfies **FAU_GEN.2 User identity association** because it ensures the ability to associate each auditable event with the identity of the user that caused the event.

This SF satisfies **FAU_SEL.1 Selective audit** because it ensures the ability to include or exclude auditable events from the set of audited events based on the defined attributes.

AU.2 Potential violation analysis and action

This SF satisfies **FAU_ARP.1 Security alarms** because it provides a function to take actions upon detection of a potential security violation.

This SF satisfies **FAU_SAA.1 Potential violation analysis** because it ensures the ability to indicate a potential violation of the TSP when the accumulation or combination of security violations reaches the predefined limit.

AU.3 Audit storage management

This SF satisfies **FAU_SAR.2 Restricted audit review** because it ensures the ability to prohibit all users read access to the audit records, except the authorized administrator.

This SF satisfies **FAU_STG.1 Protected audit trail storage** because it provides a function to protect audit records from unauthorized deletion and modification.

This SF satisfies **FAU_STG.3 Action in case of possible audit data loss** because it ensures the ability to prevent loss of audit data by informing the administrator if the audit trail exceeds predefined limit.

This SF satisfies **FAU_STG.4 Prevention of audit data loss** because it provides a function to overwrite the oldest audit records if the audit trail is full.

AU.4 Audit data reference and review

This SF satisfies **FAU_SAR.1 Audit review** because it provides the authorized administrator with a function to review audit data.

This SF satisfies **FAU_SAR.3 Selectable audit review** because it ensures the ability to search and sort audit data based on criteria with logical relations.

IA.1 Manager identification and authentication

This SF satisfies **FIA_UAU.2 User authentication before any action** because it provides a function to perform authentication of the user before allowing any other TSF-mediated actions.

This SF satisfies **FIA_UAU.7 Protected authentication feedback** because it only provides '*' or blank as authentication feedback to the Manager user while the authentication is in progress.

This SF satisfies **FIA_UID.2 User identification before any action** because it ensures the ability to identify whether to allow access before authenticating the user based on the identity of the user.

IA.2 Agent identification and authentication

This SF satisfies **FIA_AFL.1 Authentication failure handling** because it provides a function to terminate connection to Agent if 5 unsuccessful authentication attempts related to the authorized administrator's Agent authentication occur.

This SF satisfies **FIA_UAU.2 User authentication before any action** because it provides a function to perform authentication of the user before allowing any other TSF-mediated actions.

This SF satisfies **FIA_UAU.7 Protected authentication feedback** because it only provides '*' or blank as authentication feedback to the user while the authentication is in progress.

This SF satisfies **FIA_UID.2 User identification before any action** because it ensures the ability to identify whether to allow access before authenticating the administrator based on the identity of the administrator and Manager access IP.

AC.1 Reference monitor

This SF satisfies **FPT_RVM.1 Non-bypassability of the TSP** because it ensures that all system calls within the scope of control will be intercepted.

AC.2 LBAC

This SF satisfies **FDP_IFC.1 Subset information flow control** because it enforces the MAC policy when a subject accesses an object.

This SF satisfies **FDP_IFF.2 Simple security attributes** because it enforces the MAC, when a subject accesses an object, based on the label-based security attributes of the subject and object.

This SF satisfies **FDP_ITC.1 Import of user data without security attributes** because it ensures that the MAC policy is enforced when importing user data without security attributes.

This SF satisfies **FIA_USB.1 User-subject binding** because it provides a function to establish the security attributes of the subject acting on behalf of a user based on the security attributes of the user.

AC.3 RBAC

This SF satisfies **FDP_ACC.1(2) Subset access control** because it enforces the RBAC, when a subject accesses an object, based on the role-based policy.

This SF satisfies **FDP_ACF.1(2) Security attribute based access control** because it enforces the RBAC based on the role-based policy including the subject security attributes (subject identity, subject group identity, subject process) and object security attributes (operation).

AC.4 DAC

This SF satisfies **FDP_ACC.1(1) Subset access control** because it enforces the DAC, when a subject accesses an object, based on the setuid execution permission list, permitted su user list, command control list, and kill prevention process list.

This SF satisfies **FDP_ACF.1(1) Security attribute based access control** because it enforces the DAC based on the security attributes (security role status) of the subject.

SM.1 Security function initiation and stop

This SF satisfies **FMT_MOF.1 Management of security functions behavior** because it provides the authorized administrator with the ability to initiate or stop the security function.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides a function to initiate or stop security functions through the GUI and CLI.

SM.2 Security group management

This SF satisfies **FMT_MSA.1(3) Management of security attributes** because it provides the authorized administrator with the ability to manage the security group (i.e. security category), which is the non-hierarchical security attribute.

This SF satisfies **FMT_MSA.3(3) Static attribute initialization** because it provides default values for the security group (i.e. security category), which is the non-hierarchical security attribute.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the security group through the GUI and CLI.

SM.3 Security user management

This SF satisfies FIA_ATD.1 User attribute definition because it defines the security attributes of the security user.

This SF satisfies **FMT_MSA.1(3) Management of security attributes** because it provides the authorized administrator with a function to assign, change, and delete security attributes of a user.

This SF satisfies **FMT_MSA.3(3) Static attribute initialization** because it provides default values for security attributes when the authorized administrator assigns security attributes to a user.

This SF satisfies **FMT_MTD.1(2) Management of TSF data** because it provides the authorized administrator with the ability to initialize and delete the authentication data of a user when he assigns or retrieves the security attributes of the user.

This SF satisfies **FMT_REV.1(1) Revocation** because it restricts the ability to revoke security attributes of the security user to the authorized administrator.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the security attributes of the security user through the GUI and CLI.

SM.4 User role management

This SF satisfies **FMT_MSA.1(2) Management of security attributes** because it provides the authorized administrator with the ability to manage the user roles.

This SF satisfies **FMT_MSA.3(2) Static attribute initialization** because it provides default values for the user roles.

This SF satisfies **FMT_SMR.2 Restrictions on security roles** because it provides the management functions of the security group through the GUI and CLI.

This SF satisfies **FMT_SMR.2 Restrictions on security roles** because it provides a function to associate users with roles and to restrict the conditions for different roles.

SM.5 Object security attribute management

This SF satisfies **FMT_MSA.1(3) Management of security attributes** because it provides the authorized administrator with the ability to assign and delete security attributes of an object, a file.

This SF satisfies **FMT_MSA.3(3) Static attribute initialization** because it provides default values for security attributes when the authorized administrator assigns security attributes for the object, the file.

This SF satisfies **FMT_REV.1(2) Revocation** because it restricts the ability to revoke security attributes of the object, the file, to the authorized administrator.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the object, the file, through the GUI and CLI.

SM.6 Subject security attribute management

This SF satisfies **FMT_MSA.1(3) Management of security attributes** because it provides the authorized administrator with the ability to refer to the security attributes of the subject, the process.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides a function to refer to the security attributes of the subject through the GUI and CLI.

SM.7 Role-based policy management

This SF satisfies **FMT_MSA.1(2) Management of security attributes** because it provides the authorized administrator with the ability to refer to, add, modify, and delete role-based policy.

This SF satisfies **FMT_MSA.3(2) Static attribute initialization** because it provides default values when the authorized administrator adds role-based policy.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the role-based policy through the GUI and CLI.

SM.8 Allow/deny list management

This SF satisfies **FMT_MSA.1(1) Management of security attributes** because it provides the authorized administrator with the ability to refer to, add, and delete allow/deny list that is used for the DAC policy.

This SF satisfies **FMT_MSA.3(1) Static attribute initialization** because it provides default values when the authorized administrator adds allow/deny list.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the allow/deny list through the GUI and CLI.

SM.9 Audit data configuration

This SF satisfies **FMT_MTD.1(1) Management of TSF data** because it provides the administrator with the ability to configure the audit data and alarm.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the audit data and alarm through the GUI and CLI.

SM.10 Manager user management

This SF satisfies **FIA_SOS.1 Verification of secrets** because it provides a mechanism to verify that the defined quality metric is satisfied when registering the Manager user password.

This SF satisfies **FMT_MTD.1(2) Management of TSF data** because it provides the authorized administrator with the ability to delete the authentication data of a Manager user upon deletion of the Manager user.

This SF satisfies **FMT_MTD.1(3) Management of TSF data** because it restricts the ability to modify the identification and authentication data of the Manager user to the authorized administrator and allowed user.

This satisfies **FMT_REV.1(1) Revocation** because it restricts the ability to revoke security attributes of the administrator to the authorized administrator.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the Manager user through the GUI and CLI.

SM.11 Security password management

This SF satisfies **FIA_SOS.1 Verification of secrets** because it provides a mechanism to verify that the defined quality metric is satisfied when registering the security password of the SO.

This SF satisfies **FMT_MTD.1(3) Management of TSF data** because it restricts the ability to modify the authentication data of the SO to the authorized administrator and allowed user.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the Agent user through the GUI and CLI.

SM.12 Security function configuration

This SF satisfies **FMT_MTD.1(4) Management of TSF data** because it provides the authorized administrator with the ability to manage the items regarding the operational environment of security functions.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides a function to configure the operational environment of security functions through the GUI and CLI.

TP.1 Abstract machine and TSF operation testing

This SF satisfies FMT_SMF.1 Specification of management functions because it provides a function to show the results of abstract machine testing and TSF operation testing through the GUI.

This SF satisfies **FPT_AMT.1 Abstract machine testing** because it ensures that a suite of tests are performed during initial start-up of the TSF, periodically during normal operation, and at the request of the administrator to demonstrate the correct operation of the abstract machine that underlies the TSF.

This SF satisfies **FPT_TST.1 TST testing** because it ensures that a suite of self tests are performed during initial start-up, periodically during normal operation, and at the request of the administrator to demonstrate the correct operation of the TSF.

TP.2 Integrity check and management

This SF satisfies **FMT_MTD.1(4) Management of TSF data** because it provides the authorized administrator with the ability to manage the integrity check list and interval.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the execution and list of integrity check through the GUI and CLI.

This SF satisfies **FPT_TST.1 TSF testing** because it ensures that integrity check is performed for the execution files during initial start-up and periodically during normal operation to demonstrate the correct operation of the TSF and that the authorized administrator verifies the integrity of the TSF data and TSF executable code.

TA.1 Service control

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the service control policy and session control policy through the GUI and CLI.

This SF satisfies **FTA_LSA.1 Limitation on scope of selectable attributes** and **FTA_TSE.1 TOE session establishment** because it provides, for the user login service, a function to allow or deny access according to the access IP address and access time scope based on the user identity and group identity.

This SF satisfies **FMT_MCS.1 Basic limitation on multiple concurrent sessions** because it restricts the number of concurrent sessions that belong to the same user to maximum 99.

TA.2 Manager screen saver

This SF satisfies **FMT_MTD.1(4) Management of TSF data** because it provides the authorized administrator with the ability to manage the waiting time for the Manager's inactivity.

This SF satisfies **FMT_SMF.1 Specification of management functions** because it provides the management functions of the setting for waiting time for the Manager's inactivity through the GUI and CLI.

This SF satisfies **FTA_SSL.1 TSF-initiated session locking** because it locks an interactive session after time interval of the Manager user's inactivity and requires authentication of the Manager user prior to unlocking the session.

8.4.2 Rationale for the assurance measures in the TOE summary specification

The following table shows the relationship between the assurance requirements in the TOE and the assurance measures in the TOE summary specification.

[Table 8-8] Mapping TOE SARs to the assurance measures in the TSS

TOE SAR			Assurance measures in the TOE summary specification
Assurance class	Assurance component		
Configuration management (ACM)	ACM_AUT.1	Partial CM automation	Configuration Management
	ACM_CAP.4	Generation support and acceptance procedures	
	ACM_SCP.2	Problem tracking CM coverage	
Delivery and operation (ADO)	ADO_DEL.2	Detection of modification	Delivery Procedures
	ADO_IGS.1	Installation, generation, and start-up procedures	Installation Guide
Development (ADV)	ADV_FSP.2	Fully defined external interfaces	Functional Specification
	ADV_HLD.2	Security enforcing high-level design	High-level Design
	ADV_IMP.1	Subset of the implementation of the TSF	Implementation Validation
	ADV_LLD.1	Descriptive low-level design	Low-level Design
	ADV_RCR.1	Informal correspondence demonstration	Each assurance measure provides representation correspondence between the adjacent pairs among TOE Summary Specification, Functional Specification, High-level Design, Low-level Design, and Implementation Validation
	ADV_SPM.1	Informal TOE security policy model	Security Policy Modeling

Guidance documents (AGD)	AGD_ADM.1	Administrator guidance	Administration Manual
	AGD_USR.1	User guidance	(Not provided)
Life cycle support (ALC)	ALC_DVS.1	Identification of security measures	Life Cycle Support
	ALC_LCD.1	Developer defined life-cycle model	
	ALC_TAT.1	Well-defined development tools	
Tests (ATE)	ATE_COV.2	Analysis of coverage	Tests
	ATE_DPT.1	Testing: high-level design	
	ATE_FUN.1	Functional testing	
	ATE_IND.2	Independent testing – sample	N/A (Evaluator)
Vulnerability assessment (AVA)	AVA_MSU.2	Validation of analysis	Misuse analysis
	AVA_SOF.1	Strength of TOE security function evaluation	Vulnerability analysis
	AVA_VLA.2	Independent vulnerability analysis	

Configuration management

This document satisfies **ACM_AUT.1 Partial CM automation** because it states that the developer uses a CM system.

This document satisfies **ACM_CAP.4 Generation support and acceptance procedures** because it provides controls to ensure that unauthorized modifications are not made to the TOE, ensures proper functionality and use of the CM system, and is able to confirm that any creation or modification of configuration items is authorized.

This document satisfies **ACM_SCP.2 Problem tracking CM coverage** because it gives that a CM system can control changes only to those items that have been placed under CM, that placing the configuration items under CM provides assurance that they have been modified in a controlled manner, and that placing security flaws

under CM ensures that security flaw reports are not lost or forgotten and allows a developer to track security flaws to their resolution.

Delivery procedures

This document satisfies **ADO_DEL.2 Detection of modification** because it describes the requirements on the developer to detect and prevent modification to the TOE during delivery.

Installation guide

This document satisfies **ADO_IGS.1 Installation, generation, and start-up procedures** because it describes procedures necessary for the secure installation, generation, and start-up of the TOE.

Functional specification

This document satisfies **ADV_FSP.2 Fully defined external interfaces** because it describes all external interfaces of the TSF consistently using an informal style.

High-level design

This document satisfies **ADV_HLD.2 Security enforcing high-level design** because it describes the structure of the TSF in terms of subsystems and the security functionality provided by each subsystem of the TSF.

Implementation validation

This document satisfies **ADV_IMP.1 Subset of the implementation of the TSF** because it describes the implementation representation for a selected subset of the TSF consistently.

Low-level design

This document satisfies **ADV_LLD.1 Descriptive low-level design** because it describes the low-level design of the TSF in an informal style, describes the separation of the TOE into TSP-enforcing and other modules, and identifies the purpose of each TSP-enforcing module and the internal and external interfaces to the modules.

Representation correspondence

This document satisfies **ADV_RCR.1 Informal correspondence demonstration** because it provides an analysis of correspondence between the adjacent pairs among the TOE summary specification, functional specification, high-level design, low-level design, and implementation validation.

Security policy modeling

This document satisfies **ADV_SPM.1 Informal TOE security policy model** because it provides a TSP model and demonstrate correspondence between the functional specification, security policy model, and the TSPs.

Administration manual

This document satisfies **AGD_ADM.1 Administrator guidance** because it describes the administrative functions and interfaces available to the administrator of the TOE.

User guidance

AGD_USR.1 User guidance is not considered in the TOE because it does not provide security functions or interface to be used by a user.

Life cycle support

This document satisfies **ALC_DVS.1 Identification of security measures** because it describes 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.

This document satisfies **ALC_LCD.1 Developer defined life-cycle model** because it establishes and describes the model used to develop and maintain the TOE.

This document satisfies **ALC_TAT.1 Well-defined development tools** because it describes the selected implementation-dependent options of the development tools.

Test documents

This document satisfies **ATE_COV.2 Analysis of coverage** because it ensures the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

This document satisfies **ATE_DPT.1 Testing: high-level design** because it ensures that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

This document satisfies **ATE_FUN.1 Functional testing** because it consists of test plans, test procedure descriptions, expected test results and actual test results for each test.

Misuse analysis

This document satisfies **AVA_MSU.2 Validation of analysis** because it ensures that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed.

Vulnerability analysis

This document satisfies **AVA_SOF.1 Strength of TOE security function evaluation** because it shows that a strength of TOE security function analysis for each mechanism identified in the ST is performed and that the strength of TOE security function claim meets the minimum strength level defined in the PP/ST.

This document satisfies **AVA_VLA.2 Independent vulnerability analysis** because it shows, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

8.5 SOF rationale

Requirements on the strength of function apply to the password authentication mechanism used by the Agent identification and authentication function that satisfies FIA_UAU.2. The SOF claimed for the mechanism is SOF-basic.

The information to be protected by the TOE is a general data, which possesses basic level of value as assets. The threat agent is assumed to have low-level expertise, resources, and motivation.

This ST, targeting EAL4, specifies the TOE as being resistant to vulnerabilities that can be exploited by a threat agent possessing a low-level expertise, resources, and motivation.

Therefore, the SOF-Basic is sufficient to counter a threat agent possessing a low attack potential.