# Security Target for RedCastle v2.0 for Windows

| | |
|---|---|
| Doc. Ref. | RCW06D-ASE-06 |
| Doc. Ver. | Version 1.6 |
| Last Update | 2007. 06. 14. |

# Document History

| Doc. No. | Summary | Date | Author |
|---|---|---|---|
| RCW06D-ASE-01 | Initial version<br>- RedCastle v2.0 for Windows<br>- 1.1 ST Identification Modification<br>- 2.2.2 Logical Scope & Boundaries Modification<br>- 5.1 Security Function Requirements Modification<br>- 6.1 IT Security Function Modification | 2006-04-24 | S.C. Kim |
| RCW06D-ASE-02 | Minor Update<br>- 1.4 CC Conformance: Package name conforming item added<br>- 2.2.2 Logical Scope & Boundaries : Identify non-security items<br>- 5.1 TOE Security Function Requirements: FPT_ITT.1 Component Added<br>- 5.2 TOE Security Function Requirements : Apply CC Evaluation version 2.3 | 2006-05-15 | S.C. Kim |
| RCW06D-ASE-03 | Relative Contents modified according to the Protection Profile revise<br>- PP v1.1 Conformance | 2006-06-19 | S.C. Kim |
| RCW06D-ASE-04 | - 1.1 ST Identification modified<br>- 2.2.2.5 SecureOS Kernel Part Security Function modified<br>- 6.1.1.2 Security Module Scope Separation modified<br>- Other mistyping fixed | 2007-01-03 | HS Yoon |

| | | | |
|---|---|---|---|
| RCW06D-ASE-05 | - 1.4 CC Conformance : Phrase modified<br>- 2 TOE Description : Description and figure modified to be consistent<br>- SHA-1 Algorithm modified to SHA-2<br>- Open ssl-0.9.7c Version modified to openssl-0.9.8e | 2007-06-07 | HS Yoon |
| RCW06D-ASE-06 | - 2 TOE Description: Description and figure modified to be consistent<br>- Clarify Openssl-0.9.8e & version 3<br>. 3.1.2 A. SSL Protocol<br>. 6.1.7.4 Communication & Data Protection<br>. SSL(Openssl-0.9.8e) modified<br>- 5.1.2 User Data Protection<br>. FDP_ACF.1.1: Item d) deleted for its redundancy<br>. FDP_ACF.1.3: Fix wrong heading number<br>- 6.1.7.4 Communication & Data Protection Modified<br>- 7.2.2 PP Security Function Requirements added<br>: Additional FPT_ITT.1, FAU_SAA.3 Security Function Component and Contents added<br>- 7.2.3 PP Assurance Requirements added<br>: Unnecessary part deleted.<br>- [Figure 1-1] ST Document Identification<br>: TOE Identification Item deleted<br>- 5.1.2 User Data Protection: Modify 10M to 5M<br>- 6.1.5.3: Modify 50M to 5M | 2007-06-14 | HS Yoon |

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

This chapter introduces Security Target of Level-based Access Control System, RedCastle v2.0 for Windows, developed by REDGATE Co., Ltd.

## 1.1 ST Identification

This document is the security target for the CC evaluation of the RedCastle v2.0, which is the one of the access control software for the commercial operating system, and is conformant to the Common Criteria for Information Technology Security Evaluation [CC] with extensions as defined in the Label-based Access Control System Protection Profile [LSAPP].

**[Table 1-1] ST Identification**

| | |
|---|---|
| Doc. Name | RedCastle v2.0 for Windows ST |
| Doc. Version | Version 1.6 |
| Date | 2007. 06. 14. |
| Doc. Ref. No. | RCW06D-ASE-06 |
| TOE Name | RedCastle v2.0 for Windows |
| OS | Windows 2003 Server |
| CC Basis | Common Criteria for Information Security System (No.2005-25 announced by MIC, Korea) |
| PP Claims | Label-based Access Control System Protection Profile for Government v1.1, 2006-05-17 (LACSPP) |
| Assurance Level | EAL3+ |
| Author | REDGATE Co., Ltd. R&D, SC KIm |
| Key Words | Level-based Access Control, DAC, MAC |

## 1.2 Conventions and Terminology

This section contains definitions of conventions and technical terms that are used with a meaning specific to this document.

### 1.2.1 Conventions

**Iteration**

the use of a component more than once with varying operations. The result of iteration operation is marked by the serial number in a parenthesis. For examples, FAU_SAR.3(1), FAU_SAR.3(2).

**Selection**

the specification of one or more items from a list in a component. The result of selection operation is marked by the underlined italic letters.

**Refinement**

It is the addition of details to a component. The result of refinement operation is marked by the bold letters.

**Assignment**

the specification of an identified parameter in a component. The result of assignment operation is marked by the square bracket. For example, [ assignment_values ]

The identifier used in TOE security function described in summarized specification of the ST is as follows.

**Identifier of security function**

the only one identifier assigned to the security function. The usage is <function name>.# , for example, Audit.1 or Admin.1 and so on.

### 1.2.2 Terminology

**MAC, Mandatory Access Control**

A means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity.

**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.

**SOF, Strength-of-Function**

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.

**SOF-basic**

A level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential

**Iteration**

The use of a component more than once with varying operations.

### ST, Security Target

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

### Security Level

The combination of Hierarchical Classification and Non-hierarchical Category representing the sensitivity of user or information.

### PP, Protection Profile

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

### Human User

Any person interacts with the TOE.

### User

Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.

### Selection

The specification of one or more items from a list in a component.

### 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

The minimum unit of security requirements can’t be divided.

**Role**

A predefined set of rules establishing the allowed interactions between a user and the TOE.

**Operation[1]**

It is for the component to response against specific threat or satisfy specific security policy in Common Criteria (ex. Iteration, Assignment, Selection, Refinement).

**Operation[2]**

Predefined computing or work by computer commands pseudo-command

**Threat Agent**

Unauthorized user or external IT entity causing threats such as illegal access, modification, and deletion to the asset.

**External IT Entity**

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

**Authorized Administrator**

A person authorized for the TOE administration.

**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.

### DAC, Discretionary Access Control

Access control method based on the user identity or group identity.

### Assets

The information and asset protected by the security policy of TOE.

### Refinement

The addition of details to a component as the one of operations based on Common Criteria

### Security Information System Common Criteria

Common Criteria (CC), is meant to be used as the basis for evaluation of security properties of IT products and systems. Common Criteria for Information Security System was translated into Korean for CC version 2.3. It was announced by the Ministry of Information and Communication in Korea at 21 May, 2005.

### Organizational Security Policies

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

The entity executes operation in the TOE control scope

### Sensitivity Label

The security attributes represent security level of the subject or object.

### Augmentation

It is to add more than one assurance components to the Evaluation Assurance Level or Assurance Package.

### Abstract Machine

A theoretical model of a computer hardware in the software system. If TOE is application program, it will be Operating System, and if TOE is operating system, it will be firmware or hardware.

### Component

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

### Class

The collection of families which have same security objectives

### TOE, Target of Evaluation

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

### EAL, Evaluation Assurance Level

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

### Family

A grouping of components that share security objectives but may differ in emphasis or rigor.

**Assignment**

The specification of an identified parameter in a component.

**RedCastle**

Secure OS product name developed by REDGATE Co., Ltd.

**RedCastle ESM(Enterprise Security Management)**

Security management part of RedCastle. It provides manager GUI interface and performs security managing role.

**RedCastle SecureOS**

Security function processing part of RedCastle. It will reside in each server and performs access control and various security functions.

**TSF, TOE Security Function**

A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

**TSP, TOE Security Policy**

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.

## 1.3 ST Introduction

The ST overview is brief description of the TOE and detailed explanation will be given in Section 2.

The TOE is RedCastle v2.0 that is Label-based Access Control System developed by Redgate Co., Ltd.. The TOE is label-based access control system, and performs as if application program in OS. The TOE provides the following access control policy functions to prevent violation attempt against information assets reside in server system.

The TOE provides the user with multi-level based MAC(Mandatory Access Control), and the ACL(Access Control List) based DAC(Discretionary Access Control). Also, it supports the function of privileges or permission based DAC providing in OS.

The TOE consists of the RedCastle Agent and the RedCastle Manager logically.

- RedCastle Agent: RedCastle SecureOS
- RedCastle Manager: RedCastle ESM(Enterprise Security Management)

The RedCastle SecureOS (RedCastle Agent) is loaded into each OS, and is consisted of the kernel part performing MAC and DAC etc. and the application part performing other security functions. The main access control functions provided in this TOE are as follows.

- Mandatory Access Control (MAC): It provides the MAC policies based on user and multi-level security.
- Discretionary Access Control (DAC): It provides the DAC policy by ACL(Access Control List) and Privileges or Permission.

The RedCastle ESM (RedCastle Manager) is operated in Windows 2000 Professional (Service Pack 4) environment and is provided through GUI interface.

The RedCastle SecureOS is operated in Windows environment and is consisted of kernel module that performs main security functions of access control etc. and application for security administration.

All data that communicate between the security functions management part and the security administration part are encrypted through SSL(Secure Socket Layer, openssl-0.9.8e) version 3 protocol.

## 1.4 CC Conformance

This ST conforms with the following:

This ST is CC Part2 extended and Part 3 conformant, with a claimed Evaluation Assurance Level of EAL3+.

- Label-based Access Control System Protection Profile, v1.1
- Common Criteria v2.3, Part 2
- Common Criteria v2.3, Part 3
- This ST is CC Part 2 extended and Part 3 conformant, augmented by ADV_IMP.2, ADV_LLD.1, ALC_TAT.1, ATE_DPT.2, AVA_VLA.2, with a claimed Evaluation Assurance Level of EAL3+

The claimed minimum strength of function (SOF) for this TOE is SOF-medium.

## 1.5 ST Structure

The structure of this document is defined by CC Part 1.

Section 1 is the Introduction of this ST.

Section 2 is the TOE Description.

Section 3 provides the statement of TOE security environment.

Section 4 provides the statement of security objectives.

Section 5 provides the statement of IT security requirements.

Section 6 provides the TOE summary specification, which includes the detailed specification of the IT Security Functions.

Section 7 provides the Protection Profile claim.

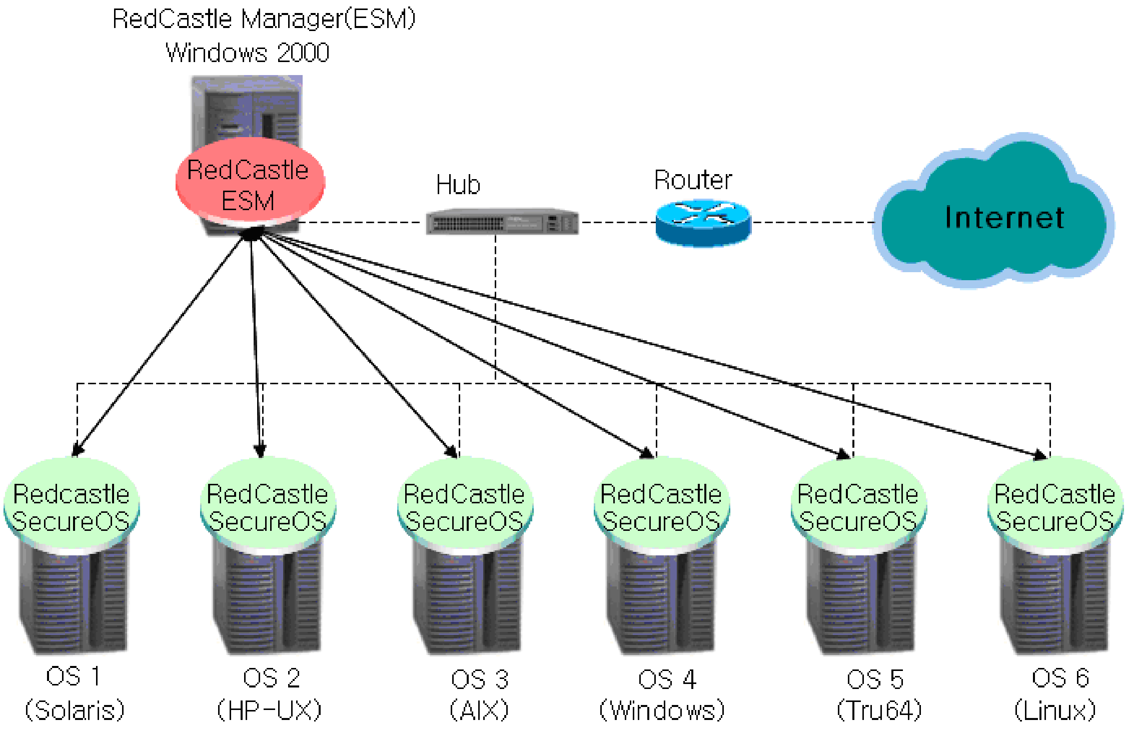
Section 8 provides the rationale for the security objectives, security requirements and the TOE summary specification.

# 2 TOE Description

This section describes the product type and TOE product configuration briefly and provides backgrounds for evaluation of TOE. The TOE description will consist of product type and TOE scope & boundaries.

## 2.1 Product Type

The RedCastle v2.0 is level-based access control system operating in OS as a program. The TOE provides MAC and DAC policies, and also supports embedded DAC policies in the OS. The TOE's brief operating configuration is as follows.



**[Figure 2-1] Typical Configuration of RedCastle**

The TOE consists of the RedCastle SecureOS which is RedCastle Agent and the RedCastle ESM (Enterprise Security Management) which is RedCastle Manager.

The main functions of RedCastle v2.0 are as follows.

- Reference Monitor
- MAC(Mandatory Access Control)
- DAC(Discretionary Access Control)
- Identification & Authentication
- Security Audit
- Security Administration
- TSF Protection

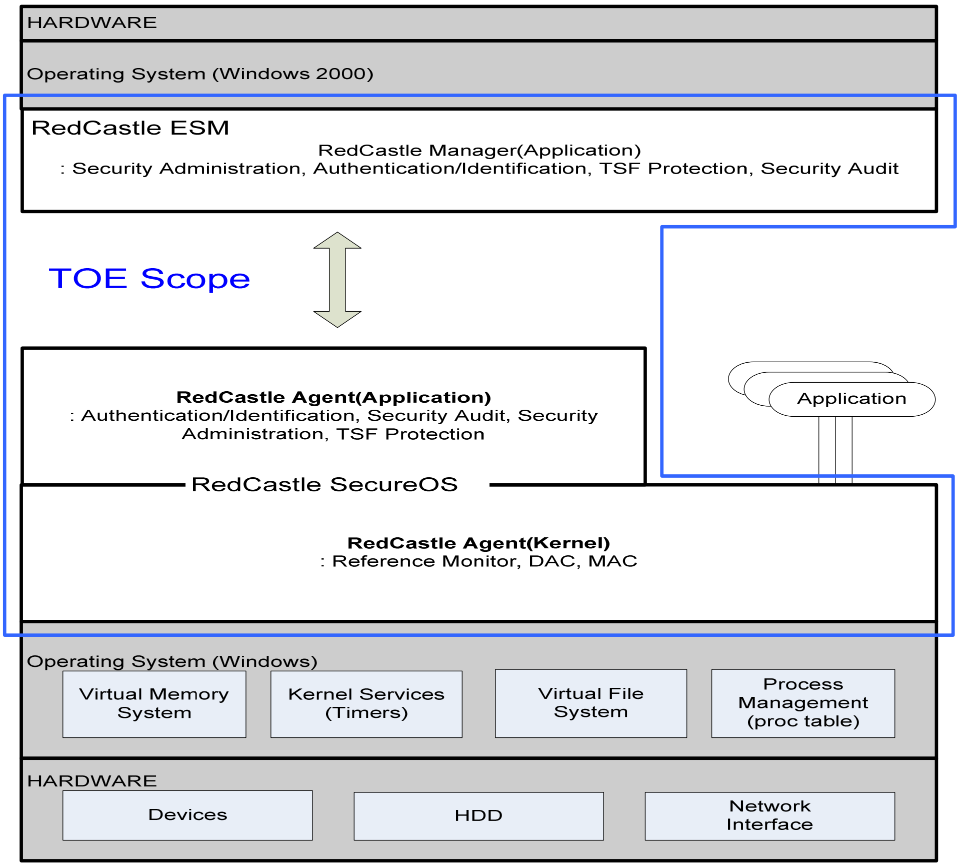
The communication between RedCastle ESM and RedCastle Secure OS uses the SSL(Secure Socket Layer) version 3 protocol to encrypt all communicating data.

## 2.2 TOE Scope and Boundaries

This section describes physical/logical scope and boundaries of TOE. Physical scope descriptions will be divided into hardware, software, and etc. The logical scope description will be about security functions of which the TOE provides.

### 2.2.1 Physical Scope and Boundaries

The TOE is RedCastle v2.0 software. The RedCastle v2.0 is divided into the RedCastle SecureOS which is RedCastle Agent and RedCastle ESM which is RedCastle Manager. The RedCastle SecureOS will be operated in application part and kernel part of OS(Windows) as a program and the RedCastle ESM will be operated in Windows 2000 Professional(Service Pack 4). The RedCastle SecureOS and the RedCastle ESM are connected through 10/100BaseT Ethernet.


HARDWARE
Operating System (Windows 2000)
RedCastle ESM
RedCastle Manager(Application)
: Security Administration, Authentication/Identification, TSF Protection, Security Audit
TOE Scope
RedCastle Agent(Application)
: Authentication/Identification, Security Audit, Security Administration, TSF Protection
Application
RedCastle SecureOS
RedCastle Agent(Kernel)
: Reference Monitor, DAC, MAC
Operating System (Windows)
Virtual Memory System
Kernel Services (Timers)
Virtual File System
Process Management (proc table)
HARDWARE
Devices
HDD
Network Interface

**[Figure 2-2] Physical TOE Scope**

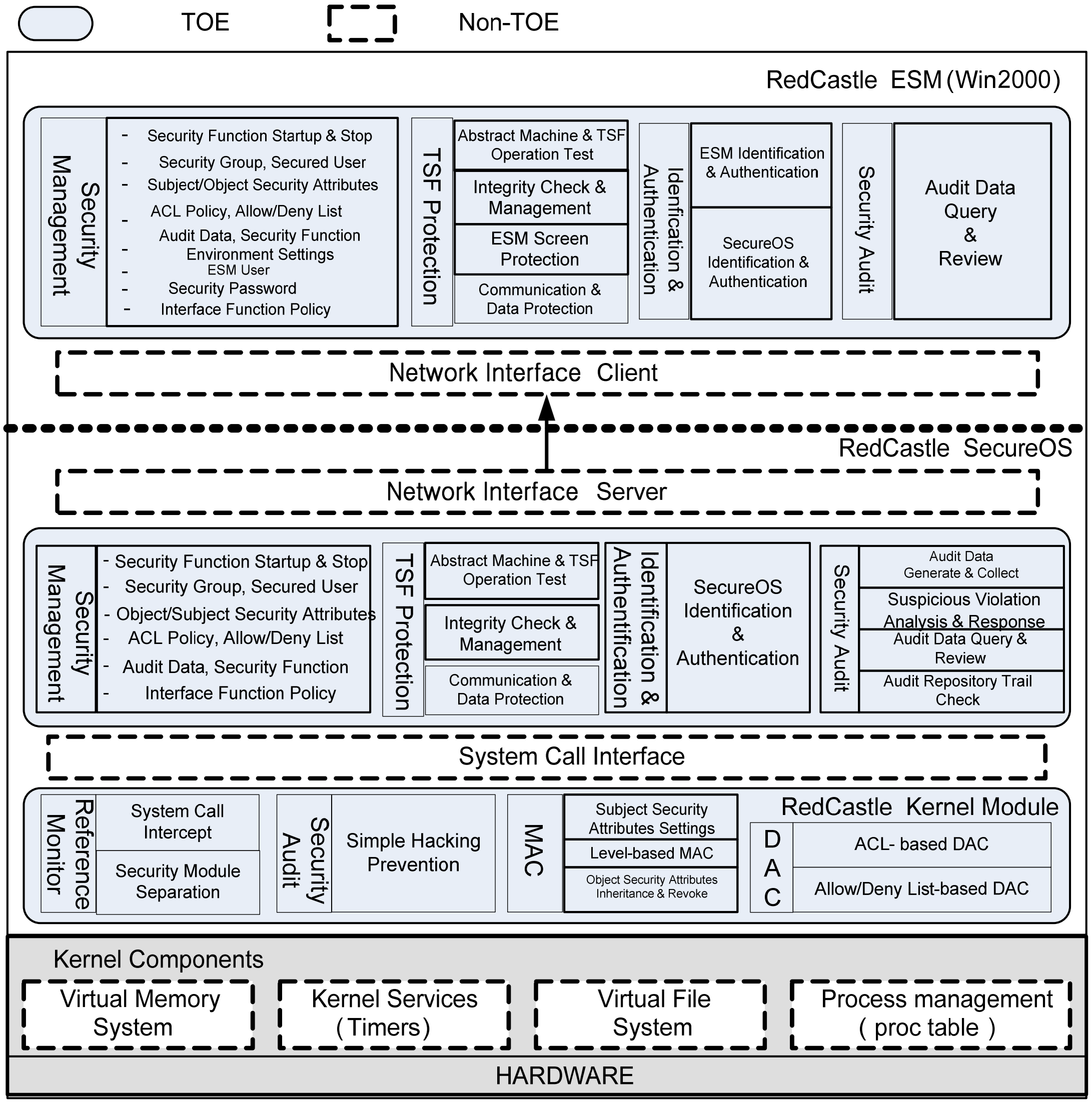
The following table is physical environment of the TOE.

**[Table 2-1] TOE Physical Configuration**

| Item | Security Function Part | Security Management Part |
|---|---|---|
| Software | RedCastle SecureOS<br>- Windows 2003 Server | RedCastle ESM<br>- Windows 2000 Professional SP4 |
| Hardware | CPU: Pentium III 600 MHz or higher<br>RAM: 256MB or higher<br>HDD: 200MB or higher<br>Network: 10/100BaseT | CPU: Pentium III 600 MHz or higher<br>RAM: 128 MB or higher<br>HDD: 10 MB or higher<br>Network: 10/100BaseT |

## 2.2.2 Logical Scope and Boundaries

The following Figure is the logical TOE scope of RedCastle v2.0.


TOE
Non-TOE
RedCastle ESM (Win2000)
Security Management
- Security Function Startup & Stop
- Security Group, Secured User
- Subject/Object Security Attributes
- ACL Policy, Allow/Deny List
- Audit Data, Security Function Environment Settings
- ESM User
- Security Password
- Interface Function Policy
TSF Protection
Abstract Machine & TSF Operation Test
Integrity Check & Management
ESM Screen Protection
Communication & Data Protection
Identification & Authentication
ESM Identification & Authentication
SecureOS Identification & Authentication
Security Audit
Audit Data Query & Review
Network Interface Client
RedCastle SecureOS
Network Interface Server
Security Management
- Security Function Startup & Stop
- Security Group, Secured User
- Object/Subject Security Attributes
- ACL Policy, Allow/Deny List
- Audit Data, Security Function
- Interface Function Policy
TSF Protection
Abstract Machine & TSF Operation Test
Integrity Check & Management
Communication & Data Protection
Identification & Authentification
SecureOS Identification & Authentication
Security Audit
Audit Data Generate & Collect
Suspicious Violation Analysis & Response
Audit Data Query & Review
Audit Repository Trail Check
System Call Interface
Reference Monitor
System Call Intercept
Security Module Separation
Security Audit
Simple Hacking Prevention
MAC
Subject Security Attributes Settings
Level-based MAC
Object Security Attributes Inheritance & Revoke
RedCastle Kernel Module
DAC
ACL- based DAC
Allow/Deny List-based DAC
Kernel Components
Virtual Memory System
Kernel Services (Timers)
Virtual File System
Process management ( proc table )
HARDWARE

**[Figure 2-3] Logical TOE Scope**

The [Figure 2-3] is logical TOE scope of RedCastle that is consisting of Reference Monitor, MAC & DAC, Security Audit, Identification & Authentication, TSF protection function such as Integrity Check, IP Filter for functioning as intrusion prevention on Network, System Account management, and Security management functions such as system monitoring.

**[Table 2-2] TOE Logical Configuration**

| Item | Security Function Part | Security Management Part |
|---|---|---|
| Security Function | RedCastle SecureOS<br>o Security Management<br>- Security function start-up & stop<br>- Security group management<br>- Secured user management<br>- Object security attributes management<br>- Subject security attributes management<br>- ACL policy management<br>- Allow/deny list management<br>- Audit data environment settings<br>- Security password management<br>- Security function environment settings<br>- Interfacing function policy management<br>o TSF Protection<br>- Abstract machine & TSF operation testing<br>- Integrity check & management<br>- Communications & data protection<br>o Identification & Authentication<br>- SecureOS Identification & Authentication<br>o Security Audit<br>- Audit data generation & collection<br>- Potential violation analysis & response<br>- Audit data repository space check<br>- Audit data query & monitoring<br><br>RedCastle Kernel Module<br>o Reference Monitor | RedCastle ESM<br>o Security Management<br>- Security function start-up & stop<br>- Security group management<br>- Secured user management<br>- Object security attributes management<br>- Subject security attributes management<br>- ACL policy management<br>- Allow/deny list management<br>- Audit data environment settings<br>- ESM user management<br>- Security password management<br>- Security function environment settings<br>- Interfacing function policy management<br>o TSF protection<br>- Abstract machine & TSF operation testing<br>- Integrity check & management<br>- ESM screen protection<br>- Communications & data protection<br>o Identification & Authentication<br>- ESM Identification & Authentication<br>- SecureOS identification & authentication<br>o Security Audit<br>- Audit data query & monitoring |

| | | |
|---|---|---|
| | - System call intercept<br>- Kernel module scope separation<br>o Security Audit<br>- Simple hacking prevention<br>o MAC<br>- Subject security attributes settings<br>- Level-based MAC<br>- Object security attributes inheritance & revocation<br>o DAC<br>- ACL-based DAC<br>- Allow/deny list based DAC | |

#### 2.2.2.1 ESM Security Function

RedCastle ESM provides the following security functions.

- ESM Identification and Authentication: Administrator must perform
- ESM administrator authentication first to utilize RedCastle's security functions through ESM. ESM provides GUI for this.
- ESM Users Management: It is to manage ESM administrator who can utilize ESM. ESM provides GUI for this.
- ESM Screen Saving: When ESM user leaves the seat for long time, it enables ESM Screen Saving. ESM provides GUI to configure waiting time for this Screen Saver function.
- Secure Communication (Client): SSL (openssl-0.9.8e) version 3 protocols are used for secure communication between ESM and Secure OS, and SSL (openssl-0.9.8e) Client will be located in the ESM.
- ESM Integrity Functions: It enables integrity check over execution files of ESM or any added files by administrator.

#### 2.2.2.2 ESM GUI

RedCastle ESM provides the following GUI for administrators to perform security functions of Secure OS.

- Security Function Start-up and Stop GUI: It enables for managing start-up and stop of security functions such as Access Control, Audit Data, and others.
- Security Group Management GUI: It enables to query, add, modify, and delete of security group (protection category) which is one of level-based security attributes.
- Secured User Management GUI: it enables to query, configure, modify, and revoke the user security attributes.
- Subject Security Attributes Management GUI: It enables to query the security attributes of process which is subject conducting instead of user.
- Object Security Attributes Management GUI: It enables to query, modify, and revoke the security attributes of file (object).
- ACL Policy Management GUI: It enables to query, add, modify, and delete the ACL-based DAC policies for file and registry.
- Allow/Deny List Management GUI: It enables to query, add, modify, and delete the Commands Control List, kill Control List, Execution Allow Command List, and Sharing Directory Access Allow List.
- Audit Configuration GUI: It enables user to configure the audit storage path, size and alarm, and the potential violation analysis.
- Security Password management GUI: It enables user to change the security password of security users.
- Interfacing Function Policy Management GUI: It provides GUI to perform functions interfacing with SecureOS.
- Integrity Management GUI: It enables user to check integrity about execution files of TSF or added data by administrator.
- Abstract Machine and TSF Operating Test GUI: It enables user to check SecureOS & system status and query audit event and potential violation analysis event in real time.

- Security Functions Configuration GUI: This function provides GUI that can configure operating environment of access control and attack prevention.
- SecureOS Authentication GUI: It enables Security Officer to connect SecureOS through ESM.
- Audit Data Query & Search GUI: It enables user to query and search the security log data managed by SecureOS and develop Audit Data Report.

#### 2.2.2.3 SecureOS Security Functions of Application Part

The application part of RedCastle SecureOS provides the following security functions.

- Secure OS Identification and Authentication: It is a function to identify and authenticate administrators to connect through ESM or security users for accessing files which have security attributes by using security password.
- Audit Generation and Collection: This function collects and stores security log, system log, and IP Filter log.
- Audit Storage Management: This function manages audit data storage and responses if it is saturated.
- Audit Review: This function handles requests of security officer (SO) to inquire and search security audit data.
- Security Functions Management: This function handles requests of security officer(SO) to start and stop security functions of access control, audit data, etc.
- Labeled Users Management: This function manages a request of security officer(SO) to inquire, configure, modify and retrieve security attributes of users.
- ACL Policies Management: This function manages a request of security officer(SO) to inquire, modify and delete ACL based policies.
- Allowed/Denied List Management: This function manages a request of security officer(SO) to inquire, add, modify and delete allowed/denied list.

- Audit Configuration: This function manages a request of security officer(SO) to configure audit data environment.
- Security Password Management: This function manages a request of security users(labeled users) to change security password.
- Abstract Machine Testing: It enables user to recognize state of Secure OS and system which operates Secure OS.
- Secure OS Integrity Functions: It enables user to check integrity about execution files of Secure OS or any added files by administrator.
- Secure Communication (Server): SSL(openssl-0.9.8e) version 3 protocols are used for secure communication between ESM and Secure OS, and SSL(openssl-0.9.8e) Client will be located in Secure OS. It also determines allowance or denial of connection to the ESM based on its IP Address and the administrator's account before the Secure OS conduct its identification and authentication process.
- Audit Data Repository Capacity Check : It monitors the capacity of audit data repository and responses for its saturation.
- Security Group Management: It handles security officer's requests for the security group such as query, add, modify, and delete.
- Subject Security Attributes Query: It handles subject security attributes query request from user.
- Object Security Attributes Management: It handles object security attributes related requests such as query, modify, and revoke from the Security Officer.
- Interfacing Function Policy Management : It handles requests for performing SecureOS interfaced functions such as IP Filter function, System Account Management function, System Monitoring function, and System Log Management function.

#### 2.2.2.4 SecureOS GUI

RedCastle SecureOS provides the Security Officer with the following GUI to perform security function of SecureOS. Some of SecureOS GUI will be provided through Operating System's console only for preventing remote control from unsecured channel.

- Security function start-up and stop GUI: It enables user to manage access control security function's start-up and stop.
- Security password GUI: It can change secured user's security password.
- Secured user authentication GUI: Through this, secured user's security password authentication is performed to access the object which has security attributes.
- Security function environment setting GUI: It enables user to configure operation environments for access control and hacking prevention function.

### 2.2.2.5 SecureOS Security Functions of Kernel Part

The kernel part of RedCastle SecureOS provides the following security functions.

- System call intercept: It replaces system call to prevent by-pass of system call by operation with loading of RedCastle kernel module in the control scope.
- Security Module Separation: Kernel module will be separated from the Operating System's list to prevent unauthorized interference from outside.
- Simple hacking prevention: It prevents simple hacking trials by hidden process detection, un-trusted program registration prevention, and registry start-up key register prevention.
- Subject security attributes settings : For subject's read and write operation on an object, access control will be enforced based on the subject and object's security levels. The subject's security attributes will be assigned when the subject is created.
- Multi-Level based MAC: It will control subject's access right to the object to conduct read and write operations based on the security levels of the subject and object. It also allocates security attributes to the subject when it is generated and inherits subject's security attributes to the object when it is generated as well as retrieve its security attributes when the object is

deleted. And if a registered program executed by an execution bypass allowing list, it will allow the execution through bypassing the multi-level based MAC and re-allocate security attributes described in the list.

- Object security attributes inheritance and revocation : It allows subject security attributes inheritance when an object is created and security attributes revocation when an object is deleted.
- ACL based DAC: ACL based DAC is applied to read, write, execute, create, delete, rename operations of a subject over an object (file and registry).
- Allow/Deny list based DAC : Allow individual operation access to the sharing directory which is registered in sharing directory access control function list. The command registered in command control function list can be executed by the security officer only. The process registered in Kill control function list can be killed by the security officer only.

# 3 TOE Security Environment

The statement of TOE security environment identifies the list of assumptions made on the operational environment and the intended method of use of the product, defines the threats that the product is designed to counter, and the organizational security policies with which the product is designed to comply.

## 3.1 Assumptions

The following conditions are assumed to be in the TOE operation environment conform with the ST.

### 3.1.1 Assumptions identical to LACSPP

IT Security Environment of this TOE has the identical assumptions to the ones of “Label-based Access Control System Protection Profile [LACSPP], v1.1.

**A. Physical Security**

It is assumed that the processing resources of the TOE will be located within controlled access facilities and will be protected from unauthorized physical modification.

**A. Trusted Administrator**

It is assumed that there are one or more competent individuals who are assigned to manage the TOE and the security of the information it contains. Such personnel are assumed not to be careless, willfully negligent or hostile.

**A. Operating System Reinforcement**

It is assumed that the operating system, where TOE is installed, is secured and reliable. Before installing the TOE, the operating system will be patched the vulnerabilities and removed the useless services.

### 3.1.2 Additional Assumptions

The following assumptions are added for this ST.

**A. SSL Protocol**

The SSL (Secure Socket Layer) version 3 protocol using openssl(openssl-0.9.8e), which is used for the secure communication between RedCastle Agent and Manager, is secured.

**A. TIME**

It is reliable for the timestamp of Operating System to be used by the TSF of this TOE.

## 3.2 Threats

The TOE must counter threats to security as below. The assumed level of expertise of the attacker for all the threats is unsophisticated, with access to standard equipment and public information.

### 3.2.1 Threats countered by the TOE

**T. CODE**

The TOE itself can be vulnerable if the developers did not develop the TOE in according to the specifications appropriately or if he includes the defect codes by accident or on purpose.

**T. AUDIT**

The auditable events of TOE, caused by the exhausting storage attack, cannot be audited.

**T. INTEGRITY**

An authorized or unauthorized user of the TOE could do the unauthorized modification or destruction of the configuration data or the sensitive information, resulting in breaking up the security functions of TOE.

**T. UAUTH**

The unauthorized authentication to the TOE could be tried by an unauthorized user or threat agent.

**T. BYPASS**

An unauthorized user may attempt to bypass the IT Environment's information flow control policy to gain access to data stored on a protected by IT system.

**T. RESIDUE**

Because an object is logically deleted (not available to the user but still within the system and may be recoverable), a threat agent could reuse the residual information illegally that is contained in a deleted object.

### 3.2.2 Threats within the TOE Environment

**TE. MISUSE**

The TOE could be configured, managed, operated by an authorized administrator.

**TE. INSTALL**

During the installation of TOE, the security of TOE could be broken off by a user who is installing.

## 3.3 Organizational Security Policies

This TOE complies with the following organizational security policies

**P. ACCOUNTABLE**

The users of the system shall be held accountability for their actions within the system.

**P. MAC**

The right to access specific labeled data objects is determined on the basis of the security label of subject.

**P. LABEL**

The TOE must assign and revoke the security label of the subject and the object according to the organization access control policies.

**P. IA**

Only users who have been through proper identification and authentication process can access to the information.

**P. ADMIN**

An authorized administrator must manage the TOE securely.

**P. CIPHER**

The encryption algorithm and its modules used for the TOE must be certified by the National Intelligence Service (NIS) in Korea.

**P. DAC**

The right to access specific data objects is determined on the basis of the identity of user or group.

# 4 Security Objectives

The statement of security objectives shall define the security objectives for the TOE and its environment. The security objectives shall address the entire security environment aspects identified.

## 4.1 Security Objectives for the TOE

The following security objectives should be handled by the TOE directly.

**O. AUDIT**

The TSF must record the security relevant actions of users of the TOE. The TOE must provide the means of investigating any security relevant events.

**O.MAC**

The TOE must enforce the access to resources on the basis of the security label of subject and object.

**O.CODE**

The source codes which are generated by developers must be inspected whether they have some defects or not.

**O.MANAGEMENT**

The TSF must provide all the functions and facilities necessary to support administrative users that are responsible for the management of TOE security and must ensure that only administrative users are able to access such functionality.

**O.INTEGRITY**

The TOE must protect the TSF data or the reliable data from the unauthorized disclosure, modification, and deletion.

**O.LABEL**

The TOE must assign and revoke the security label of the subject and the object according to the organization access control policies.

**O.IA**

The TOE must identify a user uniquely and ensure that only authorized users gain access to the TOE and its resources.

**O.DAC**

The TOE must enforce the access to resources on the basis of the identity of user or group.

**O.PROTECT**

The TOE must protect itself from the deactivation or the modification of the TOE.

**O.RESIDUE**

The TOE must ensure that any information contained in a protected resource is not released when the resource is recycled.

## 4.2 Security Objectives for the TOE Environment

All security objectives listed in this section are targeted at the non-IT environment of the TOE.

**OE.ATTACKER LEVEL**

The attacker shall have low level of expertise. Resources, and motivations and there is a little possibility to find a exploitable vulnerabilities in the system.

**OE.LOCATE**

It must be ensured that the processing resources of the TOE will be located within controlled access facilities and will be protected from unauthorized physical modification.

**OE.ADMINS**

It must be ensured that there are one or more competent individuals who are assigned to manage the TOE and the security of the information it contains. Such personnel are assumed not to be careless, willfully negligent or hostile.

**OE.MANAGE**

The TOE must be installed securely and be configured, managed, and used by an only authorized administrator.

**OE. PATCH**

It must be ensured that the operating system, where TOE is installed, is secured and reliable. Before installing the TOE, the operating system will be patched the vulnerabilities and removed the useless services.

### 4.2.1 Additional Security Objectives

All security objectives for the environment listed in this section are added in the ST.

**OE. SSL**

The TOE uses the SSL protocol for ensuring the secure communication which is provided by IT environment.

**OE. TIME**

The TOE uses the reliable timestamp which is provided by IT environment.

# 5 IT Security Requirements

This part of the ST defines the detailed functional and assurance security requirements that shall be satisfied by the TOE or its environment.

## 5.1 TOE Security Functional Requirements

All of the following security functional requirements are taken from the “Label-based Access Control System Protection Profile for Government”, version 1.1 [LACSPP].

The claimed minimum strength of function (SOF) for this TOE is SOF-medium.

[Table 5-1] shows the summary of Security Functional Components.

**[Table 5-1] Security Functional Requirements**

| Class | 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_SAA.3 | Simple attack heuristics |
| | 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 | Subset access control |
| | FDP_ACF.1 | 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 |
| | FDP_RIP.1 | Subset residual information protection |
| Identification and authentication | FIA_AFL.1 | Authentication failure handling |
| | FIA_ATD.1 | User attribute definition |
| | FIA_SOS.1 | Verification of secrets |
| | FIA_UAU.1 | Timing of authentication |
| | FIA_UAU.4 | Single-use authentication mechanisms |
| | 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 behavior |
| | FMT_MSA.1(1) | Management of security attributes (DAC) |
| | FMT_MSA.1(2) | Management of security attributes (MAC) |
| | FMT_MSA.3(1) | Static attribute initialization (DAC) |
| | FMT_MSA.3(2) | Static attribute initialization (MAC) |
| | FMT_MTD.1(1) | Management of TSF data (Audit data) |
| | FMT_MTD.1(2) | Management of TSF data (Identification and authentication data) |
| | FMT_MTD.1(3) | Management of TSF data (authentication data) |
| | FMT_MTD.1(4) | Management of TSF data (TSF data) |
| | FMT_REV.1(1) | Revocation (User) |
| | FMT_REV.1(2) | Revocation (Object) |
| | FMT_SMF.1 | Specification of management functions |
| | FMT_SMR.1 | Security roles |
| Protection of the TSF | FPT_AMT.1 | Abstract machine testing |
| | FPT_ITT.1 | Basic internal TSF data transfer protection |
| | FPT_RVM.1 | Non-bypassability of the TSP |
| | FPT_SEP.1 | TSF domain separation |
| | FPT_STM.1 | Reliable time stamps |
| | FPT_TST.1 | TSF testing |
| TOE access | FTA_SSL.1 | TSF-initiated session locking |
| Trusted path/channels | FTP_ITC.1 | Inter-TSF trusted channel |

### 5.1.1 Security Audit (FAU)

**FAU_ARP.1 Security Alarms**

Hierarchical to: No other components

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

a) [ Compulsory kill of the violated subject process

b) Alarm RedCastle ESM in real-time

c) Inform to Authorized Administrator by e-mail ]

Dependencies: FAU_SAA.1 Potential violation analysis

**FAU_GEN.1 Audit data generation**

Hierarchical to: No other components

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;

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

c) [ Successful events in case of bypass the enforcement of Mandatory Access Control,

d) Rejected events when the option of the enforcement of MAC is “Warning”,

e) Auditable Events of [Table 5-2] and [Table 5-3] ]

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, base on the auditable event definitions of the functional components included in the PP/ST, [ Auditable Events of [Table 5-3] ]

Dependencies: FPT_STM.1 Reliable time stamps

**[Table 5-2] Auditable Events**

| Component | Event | Details |
|---|---|---|
| FAU_ARP.1 | Actions taken due to imminent security violations | - |
| FAU_SAA.1 | Enabling and disabling of any of the analysis mechanisms,<br>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 | The identity of the object |
| FDP_IFF.2 | Decisions to permit requested information flows | Subject's sensitivity, label, object's name and 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 actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal) | - |
| FIA_SOS.1 | Rejection by the TSF of any tested secret | - |
| FIA_UAU.1 | All use of the authentication mechanism | - |
| FIA_UAU.4 | Attempts to reuse authentication data | - |
| 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 (e.g. creation of a subject) | - |
| FMT_MOF.1 | All modifications in the behavior of the functions in the TSF | - |
| FMT_MSA.1 | All modifications of the values of security attributes | Modified security attributes |
| FMT_MTD.1 | All modifications to the limits on TSF data | Modified TSF data value |
| FMT_REV.1 | All attempts to revoke security attributes | - |
| FMT_SMF.1 | Use of management functions | - |
| FMT_SMR.1 | Modifications to the group of users that are part of a role | - |
| FPT_STM.1 | Changes to the time | - |
| FPT_TST.1 | Execution of the TSF self tests and the result of the tests | Modified TSF data or execution code by integrity violation |
| FTA_SSL.1 | Locking of an interactive session by the session locking mechanism,<br>Successful unlocking of an interactive session | - |
| FTP_ITC.1 | Failure of the trusted channel functions,<br>Identification of the initiator and target of failed trusted channel functions | - |

**[Table 5-3] Additional Auditable Events**

| Component | Event | Details |
|---|---|---|
| FAU_SAA.3 | Enabling and disabling of any of the analysis mechanisms,<br>Automated responses performed by the tool | Identity of authorized administrator performs actions |
| FAU_STG.3 | Actions taken due to exceeding of a threshold | - |
| FAU_STG.4 | Actions taken due to the audit storage failure | - |

**FAU_GEN.2 User Identity Association**

Hierarchical to: No other components

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

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Identification

**FAU_SAA.1 Potential Violation Analysis**

Hierarchical to: No other components

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 [ Failure event among the auditable events of FIA_UAU.1, Rejected event among the auditable events of FDP_ACF.1, Violated event among the auditable events of FPT_TST.1 ] ;

b) [ Accumulation of the Simple Attack Heuristics violation ]

Dependencies: FAU_GEN.1 Audit data generation

**FAU_SAA.3 Simple Attack Heuristics**

Hierarchical to: FAU_SAA.1

FAU_SAA.3.1 The TSF shall be able to maintain an internal representation of the following signature events [ *a subset of system events as below* ] that may indicate a violation of the TSP.

a) [ Hiding backdoor process

b) An illegal practice to register hacking program such as backdoor in start-up program

c) An illegal attempts to register hacking program such as backdoor in registry start-up key

FAU_SAA.3.2 The TSF shall be able to compare the signature events against the record of system activity discernible from an examination of [ the information to be used to determine system activity ].

FAU_SAA.3.3 The TSF shall be able to indicate an imminent violation of the TSP when a system event is found to match a signature event that indicates a potential violation of the TSP.

Dependencies: No dependencies

**FAU_SAR.1 Audit Review**

Hierarchical to: No other components

FAU_SAR.1.1 The TSF shall provide [ *authorized administrator* ] with the capability to read [ *all audit information* ] 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.

Dependencies: FAU_GEN.1 Audit data generation

**FAU_SAR.2 Restricted Audit Review**

Hierarchical to: No other components

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those the authorized administrators that have been granted explicit read-access.

Dependencies: FAU_SAR.1 Audit review

**FAU_SAR.3 Selectable Audit Review**

Hierarchical to: No other components

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

a) User identity

b) Object identity

c) Security label of subject

d) Security label of object

e) Period of the audited time

f) Event type

Dependencies: FAU_SAR.1 Audit review

**FAU_SEL.1 Selective audit**

Hierarchical to: No other components

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

b) [ security label of subject

c) Security label of object

d) Period of the audited time

e) Event type ]

Dependencies: FAU_GEN.1 Audit data generation

FMT_MTD.1 Management of TSF data

### FAU_STG.1 Protected Audit Trail Storage

Hierarchical to: No other components

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 stored audit records in the audit trail.

Dependencies: FAU_GEN.1 Audit data generation

### FAU_STG.3 Action in case of possible audit data loss

Hierarchical to: No other components

FAU_STG.3.1 The TSF shall take [ the following actions ] if the audit trail exceeds [ the audit space ( = file counts x file size) defined by the authorized administrator, the default values are 5 files and 10MB per file ].

a) [ generate an alarm by e-mail to the authorized administrator if the audit trail exceeds 80% of the defined audit space.

b) Generate an alarm by e-mail to the authorized administrator for the increases at an interval of 5%.

c) If the audit trail is full, sending an alarm by e-mail to the authorized administrator and acting a function of FAU_STG.4 Prevention of audit data loss ]

Dependencies: FAU_STG.1 Protected audit trail storage

### FAU_STG.4 Prevention of Audit Data Loss

Hierarchical to: FAU_STG.3

FAU_STG.4.1 The TSF shall prevent auditable events, except those taken by the authorized user with special rights and [ take the following actions to be taken in case of audit storage failure ] if the audit trail is full.

a) [ As soon as the audit trail is full, deferring the TSF to be called by all users except the authorized administrator

b) The TSF shall be resumed by an authorized administrator after the backup of audit trail ]

Dependencies: FAU_STG.1 Protected audit trail storage

### 5.1.2 User Data Protection (FDP)

**FDP_ACC.1 Subset Access Control**

Hierarchical to: No other components

FDP_ACC.1.1 The TSF shall enforce the [ Discretionary Access Control Policy ] on [ all processes ], [ the following list of objects and the following list of operations among subjects and objects covered by SFP ].

a) [ list of objects: file system objects of the Linux Operating System

b) List of operations among subjects and objects

- read,
- write,
- execute,
- create,
- delete,
- rename ]

Dependencies: FDP_ACF.1 Security attribute based access control

**FDP_ACF.1 Security Attributes based Access Control**

Hierarchical to: No other components

FDP_ACF.1.1 The TSF shall enforce the [ Discretionary Access Control Policy ] to objects based on [ the following attributes ].

a) [ user identity associated with a subject

b) Group membership associated with a subject

c) process

d) the list of DAC attributes related the following:

- Enforce the allowance or denial of operation based on user identity
- Enforce the allowance or denial of operation based on group membership
- Enforce the allowance or denial of operation based on process
- The default value of the allowance or denial of operation is [ denial ]

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

[ rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects:

a) For each operation, if the identity of a user (subject) is specified in the attributes of the access control rules, this operation is allowed. Otherwise it is denied.

b) For each operation, if the group membership of user (subject) is specified in the attributes of the access control rules, this operation is allowed. Otherwise it is denied.

c) For each operation, if the name of process (subject) is specified in the attributes of the access control rules, the operation is allowed. Otherwise it is denied.

d) If an operation among controlled subjects and controlled objects is specified in the attributes of the access control rules, this operation is allowed. Otherwise it is denied. ]

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

[ rules, based on security attributes, that explicitly authorize access of subjects to objects

a) If the security attribute of user is an authorized administrator (Security Officer, SO), the TSF shall explicitly authorize access of subjects to objects.

b) If an operation which is called is not for an operation among controlled subjects and controlled objects, the TSF shall provide the additional rules associated with the following:

- The allowed rule for setuid operation
- The allowed rule for su operation

c) If an operation which is called is not for an operation among controlled subjects and controlled objects, the TSF shall provide the additional rules to be allowed associated with the following:

- For setuid operation, if the name of setuid program, which is executed by subject, is specified in the allowed rule of setuid operation, executing setuid program is allowed.
- For su operation, while the right of subject is escalated to root by su program, if the identity of subject is specified in the allowed rule of su operation, su is allowed. ]

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [ following rules, based on security attributes, that explicitly deny access of subjects to objects ].

a) [ If the security attribute of object is SO, the TSF shall explicitly deny access of subjects to objects except that the security attribute of subject is SO.

b) If an operation which is called is not for an operation among controlled subjects and controlled objects, the TSF shall provide the additional rules to be denied associated with the following

- The denied rule of the restricting command execution
- The denied rule of the controlling kill signal

c) If an operation which is called is not for an operation among controlled subjects and controlled objects, the TSF shall provide the additional rules to be allowed associated with the following:

- For the command execution operation, if the name of command is specified in the denied rule of the restricting command execution, executing command is denied.
- For kill operation, if the name of process is specified in the denied rule of the controlling kill signal, sending a signal to that process is denied ]

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

**FDP_IFC.1 Subset information flow control**

Hierarchical to: No other components

FDP_IFC.1.1 The TSF shall enforce the [ Mandatory Access Control Policy ] on [ all subjects ], [ list of objects and list of operations among subject and subject, and among subject and objects.

a) [ list of objects ]

- Operation among subject and subject: Subjects are user or process in the Linux operating system.
- Operation among subject and object: Objects are user, process, or file in the Linux operating system.

b) List of operations among subject and subject

- Write operation: kill

c) List of operations among subject and object

- Read operations: read, execute
- Write operations: write, delete, create

Dependencies: FDP_IFF.1 Simple security attributes

**FDP_IFF.2 Hierarchical Security Attributes**

Hierarchical to: FDP_IFF.1

FDP_IFF.2.1 The TSF shall enforce the [ Mandatory Access Control Policy ] base on the following types of subject and information security attributes:

a) [ Security Label of subject

b) Security Label of object ]

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) [ For read operation, if the security attribute of subject is greater than the one of object, a subject can read the information of object.

b) For write operation, if the security attributes of subject and object are equal, the information of subject can be written to object.

c) If the security attribute of subject A is greater than the one of subject B, the information of subject B flows to subject A. ]

FDP_IFF.2.3 The TSF shall enforce the [ additional information flow control SFP rules].

FDP_IFF.2.4 The TSF shall provide the following [ list of additional SFP capabilities ].

FDP_IFF.2.5 The TSF shall explicitly authorize an information flow based on the following rules: [ rules based on security attributes, that explicitly authorize information flows ].

a) [ provides the rules that authorize the information flows based on the security attributes of users, such as SO(Security Officer), SA(System Administrator), MU(Multi-Label Security User).

b) If the security attribute of subject is SO, the TSF shall explicitly authorize the access of subject to object.

c) If the SO executes the file which has the SA attribute, the TSF assign SA to a process.

d) The TSF shall allow executing and re-assigning the security attribute for the commands registered in the list of the command to be allowed

e) The TSF shall allow the transition user who is SO or SA to system root by the su command.

f) If the subject has the security attribute access to the non-labeled object, the TSF shall explicitly allow the access to object. ]

FDP_IFF.2.6 The TSF shall explicitly deny an information flow based on the following rules: [ rules based on security attributes, that explicitly deny information flows ].

a) [ provides the rules that deny the information flows based on the security attributes of users, such as SO, SA, MU

b) If the security attribute of object is SO, the TSF shall explicitly deny the access of subject except SO to object.

c) The SA subject cannot access to the MU object.

d) The MU subject cannot access to the SA object ]

FDP_IFF.2.7 The TSF shall enforce the following relationships for any two valid information flow control security attributes:

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

- if the security attributes are equal,
- if one security attribute is greater than the other, or
- if the security attributes are incomparable

b) There is a “least upper bound (LUB)” in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is greater than or equal to the two valid security attributes;

c) There is a “greater lower bound (GLB)”, in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is not greater than the two valid security attributes.

Dependencies: FDP_IFC.1 Subset information flow control

FMT_MSA.3 Static attribute initialization

**FDP_ITC.1 Import of user data without security attributes**

Hierarchical to: No other components

FDP_ITC.1.1 The TSF shall enforce the [ Mandatory Access Control Policy ] when importing user data, controlled under the **Mandatory Access Control 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 additional importation control rules ] when importing user data controlled under the Mandatory Access Control Policy from outside the TSC:

a) [ An authorized administrator shall specify the security attribute of the user data imported, it is based on the identity of the subject importing data.

b) An authorized administrator shall specify the security attribute of the user data imported, it is based on the security attribute of subject importing data. ]

Dependencies: [ FDP_IFC.1 Subset information flow control ]

FMT_MSA.3 Static attribute initialization

**FDP_RIP.1 Subset residual information protection**

Hierarchical to: No other components

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource from* the following objects: [ files ].

Dependencies: No dependencies

### 5.1.3 Identification and authentication (FIA)

**FIA_AFL.1 Authentication failure handling**

Hierarchical to: No other components

FIA_AFL.1.1 The TSF shall detect when *[ 5 ]* unsuccessful authentication attempts occur related to [ all authentication events ].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [ disable the account until unlocked by the authorized administrator ].

Dependencies: FIA_UAU.1 Timing of authentication

**FIA_ATD.1 User attributes definition**

Hierarchical to: No other components

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

a) [ user identifier

b) Group memberships

c) Security Label

d) Authentication Data

e) Security-relevant Roles ]

Dependencies: No dependencies

**FIA_SOS.1 Verification of secrets**

Hierarchical to: No other components

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

a) [ Password length: between 8 and 15 characters

b) Acceptable password characters:

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

c) A password must contain at least one character of alphabetic, numeric, and special characters.

d) Allow to use the consecutive alphabetic or numeric

e) Allow to use the repetition of characters ]

Dependencies: No dependencies

Application Notes: Examples of the defined quality metric could include minimum length, mixing rule, or change period for password authentication mechanism.

**FIA_UAU.1 Timing of authentication**

Hierarchical to: No other components

FIA_UAU.1.1 The TSF shall allow [ the following list of TSF mediated actions ] on behalf of the user to be performed before the user is authenticated.

a) [ Control of the Manager (RedCastle ESM) connections: whether a connection to Agent on the basis of a Manager's IP address is allow or not. ]

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

Dependencies: FIA_UID.1 Timing of identification

Application Notes: The user must be authenticated before any action and there must be no the TSF-mediated actions before authentication. But, in case of requiring the access right of an authorized administrator, the TSF-mediated actions are needed.

**FIA_UAU.4 Single-use authentication mechanisms**

Hierarchical to: No other components

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [ the following authentication mechanism ] .

a) [ Secure OS Authentication mechanism ]

Dependencies: No dependencies

**FIA_UAU.7 Protected authentication feedback**

Hierarchical to: No other components

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

Dependencies: FIA_UAU.1 Timing of authentication

**FIA_UID.2 User identification before any action**

Hierarchical to: FIA_UID.1

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.

Dependencies: No dependencies

**FIA_USB.1 User-subject binding**

Hierarchical to: No other components

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

a) [ the user identity

b) Security label

c) Security roles ]

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

a) [ The security attribute of subject acting on the behalf of users shall assign based on the following security attribute of user:

- The subject identity based on the user
- The subject security attributes based on the user
- The security role status of user identity and security roles

b) If the subject identity acting on the behalf of users would change, the role status shall be changed on the basis of the user identity. ]

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

a) [ If the security attribute acting on the behalf of users had changed, the security attribute of subject cannot be changed. ]

Dependencies: FIA_ATD.1 User attributes definition

### 5.1.4 Security Management (FMT)

**FMT_MOF.1 Management of security functions behavior**

Hierarchical to: No other components

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

a) [ Security management functions

b) Audit functions

c) Access Control functions ]

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

**FMT_MSA.1(1) Management of security attributes**

Hierarchical to: No other components

FMT_MSA.1.1 The TSF shall enforce the [DAC Policy] to restrict the ability to *change, query, and modify the default value of* the security attributes [DAC policy associated with objects] to [authorized administrator(SO), the owner of object].

Dependencies: [ FDP_ACC.1 Subset access control ]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

**FMT_MSA.1(2) Management of security attributes**

Hierarchical to: No other components

FMT_MSA.1.1 The TSF shall enforce the [ Mandatory Access Control Policy] to restrict the ability to *change, query, and modify the default value of* the security attributes [ MAC policy associated with subjects or objects ] to [ authorized administrator (SO) ].

Dependencies: [ FDP_IFC.1 Subset information flow control ]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

**FMT_MSA.3(1) Static attributes initialization**

Hierarchical to: No other components

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(SO)] to specify alternative initial values to override the default values when an object or information created.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

**FMT_MSA.3(2) Static attributes initialization**

Hierarchical to: No other components

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(SO)] to specify alternative initial values to override the default values when an object or information created.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

**FMT_MTD.1(1) Management of TSF data**

Hierarchical to: No other components

FMT_MTD.1.1 The TSF shall restrict the ability to change *default, query, delete, and clear* the [ audit data ] to [ the authorized administrator (SO) ].

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

**FMT_MTD.1(2) Management of TSF data**

Hierarchical to: No other components

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

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

**FMT_MTD.1(3) Management of TSF data**

Hierarchical to: No other components

FMT_MTD.1.1 The TSF shall restrict the ability to *modify* the [ authentication data ] to [ the authorized administrator (SO) or the owner of authentication data ].

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

**FMT_MTD.1(4) Management of TSF data**

Hierarchical to: No other components

FMT_MTD.1.1 The TSF shall restrict the ability to change *default, query, delete, clear, [ create ]* the [ TSF data associated with security ] to [ the authorized administrator (SO) ].

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

**FMT_REV.1(1) Revocation**

Hierarchical to: No other components

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 (SO) ].

FMT_REV.1.2 The TSF shall enforce the rules [ as below ]:

a) [ The authentication data associated with the security must be revoked immediately. ]

Dependencies: FMT_SMR.1 Security roles

**FMT_REV.1(2) Revocation**

Hierarchical to: No other components

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 (SO) ].

FMT_REV.1.2 The TSF shall enforce the rules [ as below ]:

a) [ The access right associated with object must be revoked immediately when the access to object is verified. ]

Dependencies: FMT_SMR.1 Security roles

**FMT_SMF.1 Specification of Management Functions**

Hierarchical to: No other components

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 the security functions

b) Management of the security category

c) Management of the security attributes of user

d) Management of the security attributes of object

e) Management of the security attributes of subject(processes)

f) Management of the ACL policies

g) Management of the allow/deny policies

h) Configuration of the audit functions

i) Management of the ESM administrators

j) Management of the Secure OS Authentication

k) Management of the file integrity functions ]

Dependencies: No dependencies

**FMT_SMR.1 Security roles**

Hierarchical to: No other components

FMT_SMR.1.1 The TSF shall maintain the roles [ as below ].

a) [ the authorized administrator (SO),

b) The owner of the authentication data ]

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

### 5.1.5 Protection of the TSF (FPT)

**FPT_AMT.1 Abstract machine testing**

Hierarchical to: No other components

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 user* to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

Dependencies: No dependencies

**FPT_ITT.1 Basic internal TSF data transfer protection**

Hierarchical to: No other components

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

Dependencies: No dependencies

**FPT_RVM.1 TSP Non-bypassability of the TSP**

Hierarchical to: No other components

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

Dependencies: No dependencies

**FPT_SEP.1 TSF domain separation**

Hierarchical to: No other components

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

Dependencies: No dependencies

**FPT_STM.1 Reliable time stamps**

Hierarchical to: No other components

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

Dependencies: No dependencies

Application Notes: This component is required only for ensuring the audit data create one after the other. So, the TOE does not implement this component but just use the time-stamp which is provided by the operating system.

**FPT_TST.1 TSF testing**

Hierarchical to: No other components

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 user* to demonstrate the correct operation of the TSF.

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

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

Dependencies: FPT_AMT.1 Abstract machine testing

### 5.1.6 TOE access (FTA)

**FTA_SSL.1 TSF-initiated session locking**

Hierarchical to: No other components

FTA_SSL.1.1 The TSF shall lock an interactive session after [ time interval of administrator inactivity ] 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: [ the authentication for the authorized administrator ]

Dependencies: FIA_UAU.1 Timing of authentication

### 5.1.7 Trusted path/channels (FTP)

**FTP_ITC.1 Inter-TSF trusted channel**

Hierarchical to: No other components

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [ the remote management functions ].

Dependencies: No dependencies

Application Notes**:** The TOE implements the FPT_ITC.1 component instead of this component.

## 5.2 TOE Security Assurance Requirements

The assurance requirements of this ST are consist of the assurance components of Common Criteria Part 3, the target evaluation assurance level for the product is EAL3+. The augmented components in this ST are listed as below.

- ADV_IMP.2: Implementation of the TSF
- ADV_LLD.1: Descriptive low-level design
- ALC_TAT.1: Well-defined development tools
- ATE_DPT.2: Testing: low-level design
- AVA_VLA.2: Independent vulnerability analysis

**[Table 5-4] Assurance Requirements**

| Class | Component | |
|---|---|---|
| Configuration management | ACM_CAP.3 | Authorization Controls |
| | ACM_SCP.1 | TOE CM coverage |
| Delivery and operation | ADO_DEL.1 | Delivery procedures |
| | ADO_IGS.1 | Installation, generation, and start-up procedures |
| Development | ADV_FSP.1 | Informal functional specification |
| | ADV_HLD.2 | Security enforcing high-level design |
| | ADV_IMP.2 | Implementation of the TSF |
| | ADV_LLD.1 | Descriptive low-level design |
| | ADV_RCR.1 | Informal correspondence demonstration |
| Guidance | AGD_ADM.1 | Admin guidance |

| | | |
|---|---|---|
| documents | AGD_USR.1 | User guidance |
| Life cycle support | ALC_DVS.1 | Identification of security measures |
| | ALC_TAT.1 | Well-defined development tools |
| Tests | ATE_COV.2 | Analysis of coverage |
| | ATE_DPT.2 | Testing: low-level design |
| | ATE_FUN.1 | Functional testing |
| | ATE_IND.2 | Independent testing - sample |
| Vulnerability assessment | AVA_MSU.1 | Examination of guidance |
| | AVA_SOF.1 | Strength of TOE security function evaluation |
| | AVA_VLA.2 | Independent vulnerability analysis |

### 5.2.1 Configuration Management

**ACM_CAP.3 Authorization Controls**

**Dependency**: ALC_DVS.1 Identification of Security Response

Developer Requirements

ACM_CAP.3.1D Developer shall provide reference for TOE.

ACM_CAP.3.2D Developer shall use Configuration Management System.

ACM_CAP.3.3D Developer shall provide CM Documents.

Assurance Requirements

ACM_CAP.3.1C Reference for the TOE must be unique for the version.

ACM_CAP.3.2C Label must be attached for the TOE references.

ACM_CAP.3.3C CM Documents must include its list and management plan.

ACM_CAP.3.4C Configuration list should have a unique identifier for its each item.

ACM_CAP.3.5C Configuration list must describe its items configuring the TOE.

ACM_CAP.3.6C CM Documents must describe the method used for identifying configuration items.

ACM_CAP.3.7C CMS must identify all configuration items in unique way.

ACM_CAP.3.8C CM plan must describe how to use the CMS.

ACM_CAP.3.9C Evidence must prove the CMS is handled by CM plan.

ACM_CAP.3.10C CM Documents must provide the evidence that shows all configuration items are managed in efficient way.

ACM_CAP.3.11C CMS must provide mechanism allows only modifications authorized in the configuration items.

Evaluator Requirements

ACM_CAP.3.1E The evaluator must confirm whether provided information is satisfying all the assurance requirements.

### ACM_SCP.1 TOE CM Coverage

**Dependency**: ACM_CAP.3 Authorization Controls

Developer Requirements

ACM_SCP.1.1D The developer must provide configuration list for the TOE.

Assurance Requirements

ACM_SCP.1.1C The configuration list must include evaluation evidence required by assurance component in the deployment presentation and ST.

Evaluator requirements

ACM_SCP.1.1E The evaluator must confirm whether provided information is satisfying all the assurance requirements.

### 5.2.2 Delivery and Operation

**ADO_DEL.1 Delivery Procedures**

**Dependency : N/A**

Developer Requirements

ADO_DEL.1.1D The developer must have documents describe the procedures how to deliver the TOE to the users.

ADO_DEL.1.2D The developer must use the delivery procedures.

Evidence Requirements

ADO_DEL.1.1C The delivery documents must describe all procedures required for security maintenance for its delivery.

Evaluation Requirements

ADO_DEL.1.1E The evaluator must confirm whether provided information is satisfying all the assurance requirements.

**ADO_IGS.1 Installation, Generation, and Start-up Procedures**

**Dependency** : AGD_ADM.1 Admin Guidance

Developer Requirements

ADO_IGS.1.1D The developer must have documents explaining all required procedures for the TOE's safe installation, initiation, and start-up.

Evidence Requirements

ADO_IGS.1.1C The installation documents must describe all required procedures for the TOE's safe installation, initiation, and start-up.

Evaluator Requirements

ADO_IGS.1.1E The evaluator must confirm whether provided information is satisfying all the assurance requirements.

ADO_IGS.1.2E The evaluator must decide whether the TOE can be configured safely by the procedures for the TOE’s installation, initiation, and start-up.

### 5.2.3 Development

**ADV_FSP.1 Informal Functional Specification**

**Dependency**: ADV_RCR.1 Informal correspondence demonstration

Developer Requirements

ADV_FSP.1.1D The developer must provide functional specification.

Evidence Requirements

ADV_FSP.1.1C The functional specification must describe the TSF and TSF external interface in informal way.

ADV_FSP.1.2C The functional specification must have internal conformity

ADV_FSP.1.3C The functional specification must describe usage objective and method of all TSF external interfaces and provide its detailed effects, exceptions, and error messages properly.

ADV_FSP.1.4C The functional specification must represent the TSF completely.

Evaluator Requirements

ADV_FSP.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

ADV_FSP.1.2E The evaluator must decide whether the functional specifications realize the TOE security function requirements precisely and completely.

**ADV_HLD.2 High-level Design Separating Security and Non-security Function**

**Dependency** : ADV_FSP.1 Informal function specifications

ADV_RCR.1 Informal correspondence demonstration

Developer Requirements

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

Evidence Requirements

ADV_HLD.2.1C High-level design should be represented in informal way.

ADV_HLD.2.2C High-level design must have internal conformity.

ADV_HLD.2.3C High-level design must describe TSF configuration with sub-systems.

ADV_HLD.2.4C High-level design must describe security functionality provided by each sub-system of the TSF.

ADV_HLD.2.5C High-level design must identify sub hardware, firmware, and software required by the TSF through representing its functions provided by supplementary protection mechanism.

ADV_HLD.2.6C High-level design must identify all interfaces in the TSF sub-system.

ADV_HLD.2.7C High-level design must identify the TSF sub-system’s external interfaces.

ADV_HLD.2.8C High-level design must describe usage objective and method of all interfaces for the TSF sub-system by providing detailed items about its effects, exceptions, and error messages.

ADV_HLD.2.9C High-level design must describe the TOE by separating TSP-performing sub-system and other sub-system.

Evaluator Requirements

ADV_HLD.2.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

ADV_HLD.2.2E The evaluator must decide whether the functional specifications realize the TOE security function requirements precisely and completely.

**ADV_IMP.2 Implementation of the TSF**

**Dependency** : ADV_LLD.1 Descriptive Low-level Design

ADV_RCR.1 Informal Correspondence Demonstration

ALC_TAT.1 Well-defined Development Tools

Developer Requirements

ADV_IMP.2.1D The developer must provide representations about whole implementation of the TSF.

Evidence Requirements

ADV_IMP.2.1C The representations of whole implementation must be defined clearly in details without anymore design procedures.

ADV_IMP.2.2C The representations of whole implementation must have internal consistency.

ADV_IMP.2.3C The representations of whole implementation must describe co-relationship of all implemented parts.

Evaluator Requirements

ADV_IMP.2.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

ADV_IMP.2.2E The evaluator must decide whether the functional specifications realize the TOE security function requirements precisely and completely.

**ADV_LLD.1 Descriptive Low-level Design**

**Dependency** : ADV_HLD.2 High-level Design Separating Security and Non-security Function

ADV_RCR.1 Informal Correspondence Demonstration

Developer Requirements

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

Evidence Requirements

ADV_LLD.1.1C The low-level design must be represented in informal way.

ADV_LLD.1.2C The low-level design must have internal consistency.

ADV_LLD.1.3C The low-level design must describe the TSF in module.

ADV_LLD.1.4C The low-level design must describe each module's objective.

ADV_LLD.1.5C The low-level design must define co-relationship between modules with provided security functions and dependencies with other modules.

ADV_LLD.1.6C The low-level design must describe the method provided by each TSP-performing function.

ADV_LLD.1.7C The low-level design must identify all interfaces of the TSF module.

ADV_LLD.1.8C The low-level design must identify external interfaces of the TSF module.

ADV_LLD.1.9C The low-level design must describe usage objective and method of all interfaces for the TSF sub-system by providing detailed items about its effects, exceptions, and error messages.

ADV_LLD.1.10C The low-level design must describe the TOE with TSP-performing module and other modules differentially.

Evaluator Requirements

ADV_LLD.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

ADV_LLD.1.2E The evaluator must decide whether the functional specifications realize the TOE security function requirements precisely and completely.

**ADV_RCR.1 Informal Correspondence Demonstration**

Dependency : None

Developer Requirements

ADV_RCR1.1D The developer must provide conformity analysis of all relative TSF representations provided.

Evidence Requirements

ADV_RCR.1.1C All related security functions in abstract TSF representations must be verified its accurate and complete realization.

Evaluator Requirements

ADV_RCR.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

### 5.2.4 Guidance

**AGD_ADM.1 Admin Guidance**

**Dependency** : ADV_FSP.1 Informal Function Specifications

Developer Requirements

AGD_ADM.1.1D The developer must provide admin guidance for system administrator.

Evidence Requirements

AGD_ADM.1.1C The admin guidance must include management functions and interfaces that can be used by the TOE administrator.

AGD_ADM.1.2C The admin guidance must describe the safe way to manage the TOE.

AGD_ADM.1.3C The admin guidance must include the functions must be handled in safe processing environment and alert for the specific privileges.

AGD_ADM.1.4C The admin guidance must describe all assumptions of user action related to safe operation of the TOE.

AGD_ADM.1.5C The admin guidance must describe all security parameters under the admin's control by representing safe value properly.

AGD_ADM.1.6C The admin guidance must describe each type of security events related to the management function which has to be performed including entity's security attributes modifications under the TSF control.

AGD_ADM.1.7C The admin guidance must be consistent with all other documents forwarded for its evaluation.

AGD_ADM.1.8C The admin guidance must describe all security requirements for administrator in IT environments.

Evaluator Requirements

AGD_ADM.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

**AGD_USR.1 User Guidance**

**Dependency** : ADV_FSP.1 Informal Functional Specification

Developer Requirements

AGD_USR.1.1D The developer must provide user guidance.

Evidence Requirements

AGD_USR.1.1C The user guidance must include all functions and interfaces can be used by the TOE users other than the administrator.

AGD_USR.1.2C The user guidance must describe TOE security functions' usage can be used by user.

AGD_USR.1.3C The user guidance must include the functions must be handled in safe processing environment and alert for the specific privileges.

AGD_USR.1.4C The user guidance must state clearly all accountabilities of user required for safe operation of the TOE including responsibility related to assumptions for user actions in the TOE security environment.

AGD_USR.1.5C The user guidance must be consistent with all other documents forwarded for its evaluation.

AGD_USR.1.6C The user guidance must describe all security requirements related to users in IT environment.

Evaluator Requirements

AGD_USR.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

### 5.2.5 Life Cycle Support

**ALC_DVS.1 Identification of Security Measures**

Dependency: None

Developer Requirements

ALC_DVS.1.1D The developer must have development security documents.

Evidence Requirements

ALC_DVS.1.1C The development security documents must include all physical, procedural, human and other security measures required to protect confidentiality and integrity of the TOE design and implementation process in development environment.

ALC_DVS.1.2C The development security documents must provide the evidence to verify reliable security measures are conformed while the TOE is developed and maintained.

Evaluator Requirements

ALC_DVS.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

ALC_DVS.1.2E The evaluator must confirm whether the proper security measures are applied.

**ALC_TAT.1 Well defined Development Tools**

**Dependency:** ADV_IMP.1 Implementation of the TSF

Developer Requirements

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

ALC_TAT.1.2D The developer must include development tools' implementation and its supplement selections in the related documents.

Evidence Requirements

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

ALC_TAT.1.2C The development tools documents must define the meaning of all commands used for implementation.

ALC_TAT.1.3C The development tools documents must define all supplementary selections for implementation clearly.

Evaluator Requirements

ALC_TAT.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

### 5.2.6 Tests

**ATE_COV.2 Analysis of Coverage**

**Dependency** : ADV_FSP.1 Informal Function Specifications

ATE_FUN.1 Functional testing

Developer Requirements

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

Evidence Requirements

ATE_COV.2.1C The analysis of test coverage must prove the conformity between TSF described in the functional specifications and test items identified in the testing documents.

ATE_COV.2.2C The analysis of test coverage must prove complete conformity between TSF described in the functional specifications and test items identified in the testing documents.

Evaluator Requirements

ATE_COV.2.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

### ATE_DPT.2 Testing : Low-level Design

**Dependency** : ADV_HLD.2 High-level Design Separating Security and

Non-security Functions

ADV_LLD.1 Descriptive Low-level Design

ATE_FUN.1 Functional Testing

Developer Requirements

ATE_DPT.2.1D The developer must provide detailed level analysis of the tests.

Evidence Requirements

ATE_DPT.2.1C The detailed level analysis of the tests must prove the test items identified in test documents are enough to prove TSF operation based on the high-level design.

Evaluator Requirements

ATE_DPT.2.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

### ATE_FUN.1 Functional Testing

Dependency : None

Developer Requirements

ATE_FUN.1.1D The developer must documentation the test results of TSF.

ATE_FUN.1.2D The developer must provide the test documents.

Evidence Requirements

ATE_FUN.1.1C The test documents must include test plan, test procedure explanation, expected test results, and actual test results.

ATE_FUN.1.2C The test documents must identify security function to be tested and describe test objectives to be performed.

ATE_FUN.1.3C The test procedure explanation must identify test items to be conducted and describe scenario to test each security function. The scenario must include dependencies on other test results.

ATE_FUN.1.4C The expected test results must verify the results can be expected by successful conduct of the test.

ATE_FUN.1.5C The test results conducted by the developer must verify all tested security functions are worked as specified.

Evaluator Requirements

ATE_FUN.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

**ATE_IND.2 Independent Testing - Sample**

**Dependency** : ADV_FSP.1 Informal Function Specifications

AGD_ADM.1 Admin Guidance

AGD_USR.1 User Guidance

ATE_FUN.1 Functional Testing

Developer Requirements

ATE_IND.2.1D The developer must provide the TOE to test.

Evidence Requirements

ATE_IND.2.1C The TOE should be in proper condition to be tested.

ATE_IND.2.2C The developer must provide the same resources with the one used in TSF functional tests.

Evaluator Requirements

ATE_IND.2.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

ATE_IND.2.2E The evaluator must test some of TSF parts to confirm whether the TOE is working as described.

ATE_IND.2.3E The evaluator must conduct sample testing by selecting some of test items to verify developer's test results.

### 5.2.7 Vulnerability Assessment

**AVA_MSU.1 Examination of Guidance**

**Dependency** : ADO_IGS.1 Installation, Generation, and Start-up Procedures

ADV_FSP.1 Informal Function Specifications

AGD_ADM.1 Admin Guidance

AGD_USR.1 User Guidance

Developer Requirements

AVA_MSU.1.1D The developer must provide the TOE Guidance.

Evidence Requirements

AVA_MSU.1.1C The Guidance (includes operations after trouble shooting) must identify all possible operation modes, its effects and related items to maintain safe operations of the TOE.

AVA_MSU.1.2C The Guidance should be complete, clear, consistent, and reasonable.

AVA_MSU.1.3C The Guidance must list up all assumptions on the intended environments.

AVA_MSU.1.4C The Guidance must list up all requirements for external security measures (includes controls for external procedures, physical location and human resources).

Evaluator Requirements

AVA_MSU.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

AVA_MSU.1.2E The evaluator must repeat all configuration and installation procedures to confirm whether the TOE can be configured and operated based on the Guidance provided.

AVA_MSU.1.3E The evaluator must decide whether all unsecured conditions can be detected by using the Guidance.

### AVA_SOF.1 Strength of TOE Security Function Evaluation

**Dependency** : ADV_FSP.1 Informal Function Specifications

ADV_HLD.1 Descriptive Low-level Design

Developer Requirements

AVA_SOF.1.1D The developer must conduct the TOE security function strength analysis of each mechanism identified in the ST.

Evidence Requirements

AVA_SOF.1.1C The TOE security function strength analysis for each mechanism must prove to satisfy or exceed the least function strength level defined in the PP and ST.

AVA_SOF.1.2C The TOE security function strength analysis for each mechanism must prove to satisfy or exceed the specified TOE security function strength level defined in the PP and ST.

Evaluator Requirements

AVA_SOF.1.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

AVA_SOF.1.2E The evaluator must confirm whether the TSF strength clarification is correct.

**AVA_VLA.2 Independent Vulnerability Analysis**

**Dependency** : ADV_FSP.1 Informal Function Specifications

ADV_HLD.2 Low-level Design separating Security and Non-security Functions

ADV_IMP.1 Partial TSF Representation of Implementation

ADV_LLD.1 Descriptive High-level Design

AGD_ADM.1 Admin Guidance

AGD_USR.1 User Guidance

Developer Requirements

AVA_VLA.2.1D The developer must conduct vulnerability analysis.

AVA_VLA.2.2D The developer must provide the vulnerability analysis documents.

Evidence Requirements

AVA_VLA.2.1C The vulnerability analysis documents must describe the TOE output analysis conducted to find the method used by a user to violate the TSP.

AVA_VLA.2.2C The vulnerability analysis documents must list up the identified vulnerabilities.

AVA_VLA.2.3C The vulnerability analysis documents must prove that all identified vulnerabilities can't be exploited in the intended environments of TOE.

AVA_VLA.2.4C The vulnerability analysis documents must justify that the TOE is immune to clear penetration attacks even though it has identified vulnerabilities.

Evaluator Requirements

AVA_VLA.2.1E The evaluator must confirm whether provided information satisfies all evidence requirements.

AVA_VLA.2.2E The evaluator must conduct penetration test based on the developer's vulnerability analysis to assure identified vulnerabilities are handled properly.

AVA_VLA.2.3E The evaluator must conduct independent vulnerability analysis.

AVA_VLA.2.4E The evaluator must conduct independent penetration test based on the independent vulnerability analysis to decide possible exploitation of additionally identified vulnerabilities through intended environment. .

AVA_VLA.2.5E The evaluator must decide whether the TOE has immunity to the penetration attack which has low success rate.

## 5.3 Security Requirements for the IT Environment

The security requirements for the IT environment are as below.

**FPT_ITT.1 Basic internal TSF data transfer protection**

Hierarchical to: No other components

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

Dependency: None

**FTP_ITC.1 Inter-TSF trusted channel**

Hierarchical to: No other components

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [ the remote management functions ].

Dependencies: No dependencies

**FPT_STM.1 Reliable time stamps**

Hierarchical to: No other components

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

Dependency: None

## 5.4 Strength of Function

The TOE has the password based security function for identification and authentication that is implemented by a probabilistic or permutation mechanism. The mechanism rated in the strength of function analysis is the password mechanism for user authentication. The strength claimed for this function is SOF-medium. The security functional requirements to be claimed the SOF-medium are the FIA_UAU.1 and FIA_UAU.4, and the security functions of this TOE to satisfy the SOF-medium are Auth.1, Auth.2 and Protect.4.

# 6 TOE Summary Specification

## 6.1 TOE Security Functions

All of the following security functional requirements are taken from the “Label-based Access Control System Protection Profile for Government”, version 1.1 [LACSPP]. The claimed minimum strength of function (SOF) for this TOE is SOF-medium.

### 6.1.1 Reference Monitor

#### 6.1.1.1 Reference Monitor (Refer.1)

TSF provides the reference monitor function to ensure the calling and succeeding of TSP enforcing functions before each function in TSC ( TSF Scope of Control ) to be allowed its conduct.

In the TOE, the reference monitor function is provided with the following purposes.

- TSF provides the reference monitor function to ensure the calling and succeeding of TSP enforcing functions before each function in TSC (TSF Scope of Control) to be allowed its conduct.
- For the security policy and security function environment settings of RedCastle Secure OS, system call will be added for use in the TOE. The newly added system call will be used to collect audit record also in the audit recording subsystem.

Reference monitor function in the TOE performs as follows.

- Control system call to control security function will be registered when RedCastle kernel module is loaded into the operating system.
- Security management function command requested by authorized admin will be intercepted through reference monitor and conduct security functions such as kernel module security function starts and security policy management.

- RedCastle filter driver will be loaded into the operating system's file driver through AC security function start-up command requested by an authorized admin.
- The system call in the TOE control scope requested by any applications will be intercepted by reference monitor and controlled by access control function.

Security Requirements to be satisfied: FPT_RVM.1

#### 6.1.1.2 Security Module Separation (Refer.2)

By deleting the security module from the operating system's kernel module list when the security module is loaded into kernel, the TSF will be protected from external interference and tampering by subjects that are not trusted.

This function is not provided with security module information not only to a unauthorized user but also to a security officer(SO), and SO can confirm loading availability and performing status of security module by GUI(Graphical User Interface) or CLI(Command Line Interface).

Security Requirements to be satisfied: FPT_SEP.1

### 6.1.2 Mandatory Access Control (Ac_mac)

#### 6.1.2.1 Security Label Assignment (Ac_mac.1)

Before the TOE performs access control function, identification and security attribute assignment to the subject will be conducted first. As the information of subject in general LINUX system can be changed by su and setuid program, assignment security attribute to a subject based on its information can not be coherent when it tried at a time of applying mandatory access control (MAC). Therefore, to grant MAC policy, the information of subject should be kept coherently.

Before apply access control rule, the TSF must know identification and security attribution of subject and object. Therefore, in the subject security attribute assignment function, it tries to identify system information and security attribute of subject and object and if it is not identified properly, it can arrange security attributes according to the subject's own character.

In this TOE, if a user does login or a new process is created, the procedure of subject identification and user's security attributes-subject connection will be performed as follows before the TSF allows any actions concerned.

- After user login, the security attribute assigned to the user will not be assigned to the subject before the independent identification and authentication process conducted on it.
- The identified and authenticated subject will be assigned the user's security attribute, and after that point, all the created new processes will inherit security attributes of parent processes.
- The subject security attribute that is assigned to a process will be presented in [Role status, Security level, Security category] and each item is as follows.
    - Security label of subject
        - Security Level
        - Security Category
    - Role status(Process status) of subject

**[Table 6-1] User Security Role**

| Security Role | ID | Description |
|---|---|---|
| Security Officer | SO(3) | The user authorized as a security officer to perform the highest security officer role. (Always belonged to the Security Officer Group) |
| System Admin | SA(2) | The user authorized as a system admin to conduct the applied service and system management. (Who has Security ID of 2 or belonged to the System Admin Group) |
| Secured User | MU(1) | The secured user who has security attributes to handle classified information. (User belonged to the Security Officer Group or Secured User |

| | | |
|---|---|---|
| | | Group ; who doesn't have security level of 0) |
| System User | UX(0) | The user who doesn't have the security attributes. (Security level and security group should be 0) |
| SYSTEM TOKEN | SYSTEM | In Windows, there is a SYSTEM Token which is used with the concept of 'internal user' in the operating system other than the user. It will be used by operating system to access subsystem or kernel for operation. |

The process in Windows OS will have different properties based on how it generated. The TOE classifies process in Widows OS into the following three kinds and manages by assigning security attributes to the subject process.

- SYSTEM Process: It is the highest process in Windows and all other processes will be created under it. In the TOE, if the processes perform subsystem functions of Windows, it will be classified as system process even if those are under the system process.
- SERVICE Process: Those are processes started by operation of Windows server such as IIS Web Service, FTP Service, and Mail Service. Those processes can designate a user to start the service and in the TOE, its security attributes will be assigned based on the user when the process is started to run. If it is a service started by Administrator, it will have Administrator's security attributes (UX, 0, 0).
- USER Process: If a user login Windows system, explorer.exe process called as background view (equivalent to 'shell' in UNIX) will be executed by login user's authority. In the TOE, those all sub processes created under explorer.exe will be classified as 'USER process' and will have security attributes based on the user. The Access Control function of the TOE will be applied basically when a User Process accesses an object.

A security role status is one of security attributes should be assigned based on user's identity and security role when a user connected to as a subject. All subjects must have one of the following values

[Table 6-2] Subject Security Role Status

| User ID | User Role | Role Status | Description |
|---|---|---|---|
| user | SO | SO(3) | Subject created by a user belonged to the security officer group |
| user | SA | SA(2) | Subject created by a user belonged to the system admin group |
| user | MU | MU(1) | Subject created by a user belonged to the secured user group |
| user | UX | UX(0) | Subject created by a user who doesn't have the security attributes |
| system | SYSTEM | - | SYSTEM process |

Security attributes assignment and inheritance rule in the TOE based on the system condition can be classified as follows.

- When a new process is created, if the upper process has security attributes then the security attributes will be inherited to the lower process.
- When a user login, the subject process will have security attributes of (UX, 0, 0) until the user get through the identification and authentication process provided by system and performs separate SecureOS identification and authentication process.
- If the identification and authentication through security password has done, the security officer will assign the same security attributes assigned to the subject process owner. For example, if a security officer login with the security attributes (UX, 0, 0) and get through the identification and authentication process by security password, the subject process will have new security attributes (SO, 1, 1).
- SYSTEM process, the highest process in Windows OS, will be identified separately and all accesses to objects will be allowed for the operating system's normal operation.

- Security attributes of process which is implemented automatically with SERVICE process will be assigned based on the security attributes of the user who runs the service.

To accomplish the above mentioned rules, the TOE will apply the following security attributes assignment rules.

- SYSTEM process, the highest process, does not have security attributes.. SYSTEM process will be allowed all accesses to objects.
- SERVICE process of Windows server will have security attributes based on the security attributes of the user running service. For example, IIS Web Service inetinfo.exe will execute its service with IUSR_hostname account, if the security attributes of IUSR_hostname is (SA, 1, 1), the security attributes of inetinfo.exe will be (SA, 1, 1) too.
- If a user login to Windows server through console or Terminal Service, all processes of user will be created under Explorer.exe.
- If the identification and authentication through security password is successful after a user login, the security attributes assigned to the corresponding user will be assigned to the subject.
- Even in ftp case, the security attributes of (UX,0,0) will be assigned until the identification and authentication through security password is done successfully.

Satisfying security function requirements: FDP_IFC.1, FDP_IFF.2, FIA_USB.1

#### 6.1.2.2 Multi-Level based MAC (Ac_mac.2)

In subject security attributes setting function, level-based MAC function will be conducted after all processes are identified and security attributes are assigned.

MAC is an access restricting method to an object based on the sensitivity (allowed level) of information include in the object and clearance which is an authority owned by subject for the sensitivity accessing information.

MAC policy for access control applies a lot of information required for strong protection between the classified system data and users in each level. MAC also can be defined as an Information Flow-control policy for it is preventing information flow by the object with low security level. The access to the data will be determined by compulsory policy through definition of security level owned by subject and object.

The TOE enforces level-based MAC based on subject and object security attributes type as follows.

- Security Label of Subject
- Security Label of Object

The security label of subject and object will be defined as follows.

- Category
- Classification

In the TOE, the following relationship will be established for the above two sensitivity labels.

- The ordering function to decide the following for the above mentioned sensitivity labels will exist.
    - One sensitivity label is same with the other one
    - One sensitivity label is bigger than the other one
    - It is impossible to compare two sensitivity label

In the TOE, modified B&L(Bell & La Padula) model is used for level-based access control model. Generally, the B&L model can be summarized as the following two basic principles.

- No write-down secrecy: If the sensitivity label of object is superior than the subject's, write can be done on the object.

- No read-up secrecy: If the sensitivity label of subject is superior than the object's, it can read the object.

When a user accesses to read a file:

Subject(Security level, Category) ≥ Object(Security level, Category)

It means as Subject(Security level) ≥ Object(Security level) and Subject(Category) $\supseteq$ Object(Category). In other words, read will be allowed when a subject's security level and category is higher than or same with the file's security level and category. It also means that even if high leveled user can not modify the low-leveled file other than just read it.

When a user accesses to write a file:

Subject (Security level, Category) = Object (Security level, Category)

It means as Subject (Security level) = Object(Security level) and Subject(Category) = Object(Category). In other words, write access can be allowed only when the user's security level and category is same with the file's security level and category.

As the modified B&L model in the above mentioned, the TOE will keep the following rules based on the order between security attributes and allows information flow between controlled subject and information through controlled operation.

- If a subject's label is higher than or same with the object's label, the subject can read the object.
- If a subject's label is same with the object's label, the subject can write on the object.
- If the label of subject A is higher than or same with the label of subject B, information in the subject B can flow to the subject A.

In the Read and write rules used in the TOE's level-based access control, the following operation will be applied.

- Read rule : read, execute operation

- Write rule : write, create, delete operation

To enforce level-based access control policy effectively in the TOE, the security role status of subject representing the authority of security category will be compared with security role of an object instead of comparing level of both security categories. The TOE's level-based access control compulsory rules will be conducted as follows.

- The subject with security attributes of (SO, 1, 1) will be allowed its access to all objects and a subject other than security officer group can't access to the object (security role SO) of security officer group. In the MAC between the subject and object of security officer group, the access allowance will be determined by comparing of security level.
- The subject of system administrator group (security role SA) can't access to the object of secured user group (security role MU).
- The subject of secured user (security role MU) can't access to the object of system administrator group (security role SA).
- When a subject of system administrator group (security role SA) accesses to an object of system administrator group (security role SA), the access allowance will be determined by comparing the level of subject and object.
- When a subject of secured user group (security role MU) accesses to an object of secured user group (security role MU), the access allowance will be determined by comparing the level of subject and object.
- If the object doesn't have security level (security role UX), it will be defined as the lowest security attributes in MAC.

In the TOE, for data inbound and outbound such as FTP, the above mentioned same rule will be applied without specific function. Data outbound in FTP is equivalent to read rule in system call and data inbound is equivalent to create operation in write rule. Therefore, in case of data inbound, in-bounded data's sensitivity label will be clarified based on subject's sensitivity label.

The TOE does not enforce MAC rule for additional information flow (Biba, Lattices model, etc.).

Satisfying security function requirements: FDP_IFC.1, FDP_IFF.2, FDP_ITC.1

#### 6.1.2.3 Inheritance and Revocation (Ac_mac.3)

When an object is created, the object will inherit the subject's security attribute automatically and when it is deleted, the security attribute and information of object will be deleted through the function of object security attribute inheritance and revocation. This function assures the prevention of previous information abuse.

- ➢ If a subject generates an object, the subject security attributes will be inherited into the object.
- ➢ When a subject reads an object and if its contents copied to the other object, the original object's security attributes will be inherited to the other object.
- ➢ When a subject deletes an object (ex, revoke resource from a file), all security attributes and contents of the file will be revoked to assure all previous resources are not available any more,

Security Requirements to be satisfied: FDP_ITC.1, FDP_RIP.1

### 6.1.3 Discretionary Access Control (Ac_dac)

#### 6.1.3.1 ACL based DAC (Ac_dac.1)

System call which allowed in MAC is transmitted to DAC. DAC forces access control rules on the basis of ACL(Access Control List) according to identity of subject in DAC.

This TOE provides DAC on the basis of ACL(Access Control List) separately with DAC by permission bit supporting in OS.

Subject information that can be identified on ACL based DAC policy is as follows.

- ➢ User identifier
  - User ID
  - User name

- Group memberships
    - Security category
    - Security-relevant Roles
- Subject program name
    - Process name

ACL based DAC control the following operation.

- read
- write
- execute
- create
- delete
- rename

The following rule will be enforced to decide whether an operation will be allowed between controlled subject and object in the TOE.

- The identity of subject user or group for each operation should be the same with the identity of user or group specified in the object's access control attributes.
- If a specific program is designated, the operation will be allowed only through the allowed program.
- The access will be allowed for the selected operation among controlled operations.
- Other operations except the controlled operations will not be enforced.

Security Requirements to be satisfied: FDP_ACC.1, FDP_ACF.1

#### 6.1.3.2 Allowed/Denied List based DAC (Ac_dac.2)

Allowed/Denied list based DAC is control according to allowed/denied list for operation to manage specially or for operation that is not applied to ACL.

Allowed/Denied list is separated as follows.

- ➢ Policy based allowed list: If an act performed that is not in the allowed list, it will be denied.
    - Policy based denied list: If a user performs an act which is in the denied list, it will be denied unless a user is SO.

The TOE uses subject’s user identity for security attributes and the rules based on the following list will be enforced to determine whether the operation between the controlled subject and objects will be allowed.

- ➢ Command control list: The execution of commands registered on the command control list will be allowed only to the security officer and other users can’t be allowed.
- ➢ Kill prevention process list: The kill of processes registered on the kill prevention process list will be allowed only to the security officer and other users can’t be allowed.
- ➢ Sharing directory access allow list: Only for the registered Windows sharing directories in the list can be allowed for the specific user through Windows login and the sharing directory which is not registered or unregistered user’s accesses are not allowed.

The system call allowed in DAC will be returned to the reference monitor and DAC function provided in commercial operating system will be conducted as the original system call is called.

Satisfying Security Function Requirement: FDP_ACC.1, FDP_ACF.1

### 6.1.4 Identification and Authentication (Auth)

There are three different kinds of Identification and authentication in the TOE ; user's identification and authentication to use ESM, SO's identification and authentication for connecting to SecureOS by ESM, and identification and authentication of labeled users to gain permission before access to file of security attribute.

#### 6.1.4.1 ESM Authentication (Auth.1)

RedCastle ESM's user will be classified as follows.

- ESM administrator: A user which can connect to RedCastle Secure OS. It can add or delete ESM user.
- ESM user: A user who can be connected to RedCastle SecureOS.

ESM administrator and user (ESM user hereinafter) can be connected to the RedCastle Secure OS through the ESM identification and authentication in the system that RedCastle ESM is installed.

In the RedCastle ESM's identification and authentication function, the access right to ESM will be decided through the authentication data verification by decryption of encrypted authentication data which is generated when an ESM user was registered based on the authentication information provided by an ESM user.

If five unsuccessful authentication attempts were occurred, repetition will be prohibited because the RedCastle ESM is killed compulsory.

Satisfying Security Function Requirement: FIA_AFL.1, FIA_UAU.1, FIA_UAU.7

#### 6.1.4.2 Secure OS Authentication (Auth.2)

The Secure OS Authentication provides following security function.

- SecureOS identification and authentication of SO by ESM.
- Identification and authentication of SecureOS user(SO, SA, MU).

(1) SecureOS identification and authentication through ESM

ESM user can connect by each server that RedCastle SecureOS is installed if succeed in login to console for administration in RedCastle ESM. SO must identify and authenticate about relevant SecureOS in case of connect to each server.

Before perform SecureOS identification and authentication through ESM, it provides the identifying functions whether administrator account and ESM IP can connect to the Secure OS.

Identification and authentication information, that is provided by SO through GUI for Secure OS identification and authentication, are as follows.

- System account(Security Officer ID)
- System account password
- Secure OS security password

The communication between RedCastle SecureOS and RedCastle ESM uses the SSL(Secure Socket Layer) version 3 protocol. The SSL protocol encodes data to secure the authentication information.

(2) Secure OS identification and authentication of user for login

All users identified in Secure OS will be permitted the access to the object which has its own security attributes after the completion of authentication process by security password. Users who must through the identification and authentication process are as follows and the identification and authentication process for system root and system user who has security level ‘0’ will be rejected.

- Security Officer (SO)
- System Administrator (SA)
- Multi-Label Security User (MU)

This TOE provides CLI(Command Line Interface) for login user's identification and authentication process, and the information provided through CLI is as follows.

- User’s security password

Security Requirements to be satisfied: FIA_AFL.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.7, FIA_UID.2

The system account and password means operating system’s account registered in the RedCastle SecureOS installed system and the corresponding account should be included in the security officer group (security group ID is 1) for successful identification and authentication as security officer.

In the TOE, the identification and authentication of RedCastle SecureOS will be conducted as follows.

- The user got through the ESM identification and authentication enters system account registered as security officer, system & security password.
- The password will be displayed as ‘*’ to protect authentication feedback when it is entered.
- SecureOS will identify system account and ESM connection IP first to check whether the account and IP are connectable to SecureOS.
- Authenticate the password of OS matches with the corresponding account.
- Identify the ID provided by user whether it is included in security officer group.
- Decrypt the encrypted authentication data stored in security password file based on ID and password.
- Check whether there is a value matching with user ID in the encrypted authentication data.
- If there is a matching value, it means the identification and authentication process is succeeded by authenticating user entered ID and password.

If the authentication failure limit times (default 5 times) which is set for the individual user is reached, the connection to RedCastle SecureOS will be off and the authentication trial of the user will be denied until the security officer clear it. And an audit data for this authentication denial will be generated and sent to the registered e-mail address. If the security officer is prohibited for authentication, the other security officer can clear it through the GUI provided by ESM or the security officer can clear it by himself through system console login and successful security password authentication.

There are the following connection modes between the ESM and SecureOS.

- Management Mode: It allows the security officer to do security management activities such as security policy development and all other security functions such as log query.
- Query Mode: It allows the security officer to do real time based log query and query provides log search function only.

The management mode is allowed only one connection and query mode is allowed multiple connections. Only a user with security attributes of (SO, 1, 1) can connect in management mode and a user with security attributes of (SO, 2 - 7, 1) – security officer who doesn't have security level of 1, can connect in query mode only. When a user with security attributes of (SO, 1, 1) tries to connect in management mode, if the management mode session is connected with the same ID, the user can terminate the existing session compulsively and connect in the management mode. If there is a management session connected with other ID, the user can connect in query mode.

For the communication from RedCastle ESM to RedCastle SecureOS, SSL(Secure Socket Layer, openssl-0.9.8e) version 3 protocol will be used. For key exchange and authentication method, anonymous authentication mechanism (anonymous DH) is used and as a cipher-suite, AES encryption algorithm (256 bit encryption key) and SHA Hash algorithm (ADH-AES256-SHA) is selected. Since the SSL(openssl-0.9.8e) version 3 protocol encrypts transmitting data by creating session key randomly, the authentication information will be protected by SSL(openssl-0.9.8e) version 3 protocol and reuse of the authenticated data is prevented.

Satisfying security function requirements : FIA_AFL.1, FIA_UAU.1, FIA_UAU.4, FIA_UAU.7, FIA_UID.2

### 6.1.5 Security Audit (Audit)

#### 6.1.5.1 Audit Generation and Collection (Audit.1)

The TOE generates the following audit data and it will be collected by log daemon and saved.

- Security log generated through security management function
- Kernel log generated from RedCastle kernel module
- Log generated from system monitoring function
- Log generated from IP Filter product

Also, The TOE's log daemon provides the function to collect system log located in the outside of the TOE and sore it in the specified position separately.

- Windows event log : Application
- Windows event log : Security
- Windows event log : System

In the security management function and kernel module security function of the TOE generate security log for the following audit target events.

- Counter measure action details for potential security violation detected
  - Subject process compulsory shut-down details
  - E-mail sending details to admin
- Start-up and shut-down of audit function
- Modification details of security audit environmental settings
- Security log backup and initialization details
- Operation control status of DAC function
- Operation control status of MAC function
- Detection status of hacking prevention violated events
- Security module protection and revocation status
- Execution details of Execution allowed commands

- Success and fail history of SecureOS identification and authentication
- Authentication failure threshold reach history and associated responses
- Authentication prevention release history of security office
- ESM connection IP controlled details
- Attributes modification history of security group and secured user
- Object security attributes modification history
- Communication session fault history between ESM and SecureOS
- Accumulation violation detect and responses for access control violation log
- Audit data storage saturation alert and responses details
- Block history of unsecured ESM connection trials
- Add, modify, and delete history of ACL-based DAC policy
- Add, modify, and delete history of MAC policy
- Add, modify, and delete history of allow/deny list
- Security function environment setting status
- Integrity target list managing history and integrity check details
- Start-up and shutdown history of access control security function
- Start-up and shutdown of IP Filter function
- Add, modify, and delete history of IP Filter policy
- Add, modify, and delete history of system account/group
- Success and fail history of ESM identification & authentication
- ESM user register, delete and password modification history
- ESM screen lock setting history
- ESM screen lock function operation history
- Success and fail history of ESM screen lock release

Each security log will consist audit data including the following items.

- Audit occurring time

- Audit occurring location: Audit data generated location (communication daemon, log daemon, kernel module, etc.).
- Alert level
  - Information: Normal audit level – Successful policy modification, security function start-up and shutdown, etc.
  - Notice: Security violation events such as authentication failure, access control violation, etc.
  - Warning: Events classified as potential violation by accumulation of notice level events.
  - Critical: Serious events causing shutdown of TOE operation.
  - Error: Event such as failure of policy modification trial by security officer.
- Audit message

The access control security violation log of RedCastle kernel module will configure audit data for the following additional items and will be stored in security log file.

- Subject information
  - User ID (User Name)
  - Process ID
  - Security group of process
  - Security level of process
  - Role status of process
- Object information
  - File, directory name, registry
  - Security group of object
  - Security level of object
- Occurring operation : System call information
- Violation message

Satisfying security function requirement: FAU_GEN.1, FAU_GEN.2, FAU_SEL.1, FPT_STM.1

#### 6.1.5.2 Potential Violation Analysis (Audit.2)

The SO can define unit time and the accumulation frequency limit for potential violation analysis in collected audit data in the TOE. In collected audit data, if a reviewed security violation exceeds accumulation frequency per unit time, TSF detects and warns this.

Rules for potential security violation analysis are as follows.

- Unit time of potential security violation analysis
- limit frequency

A set of rules for potential security violation analysis are as follows.

- Violation of identification and authentication security policy
- Violation of access control rules
- Violation of other security policy

Satisfying security function requirement: FAU_ARP.1, FAU_SAA.1

#### 6.1.5.3 Audit Storage Management (Audit.3)

Audit storage provides 50MB of file size and 5 files count in default value, and if this limit is exceeded, it will warn the administrator by registry e-mail, and perform audit data loss prevention function.

The TOE provides function for SO to configure audit storage environment.

Security Requirements to be satisfied: FAU_STG.1, FAU_STG.3, FAU_STG.4

#### 6.1.5.4 Audit Review (Audit.4)

In this TOE, the audit record will be stored in the directory to which SO can access only and provided in the format of which SO is able to read. The audit data which can query and review in this TOE is as follows.

- Security log
- System log
- System monitoring log
- IP Filter log

SO queries and reviews a selective audit data through GUI provided in RedCastle ESM.

The SO can develop the report about the stored audit data by using GUI(Graphical User Interface).

- Audit occurred time
- Audit occurred location
- Alarm level

Access control violation log of kernel module can be searched and sorted by the following items.

- Audit occurred time
- Audit target event
- Subject information
  - Process name
  - Process owner name
  - Role status
  - Security group
- Object information
  - File, registry, target process name
  - Security group

- Security role

The security officer can generate the following report for the stored audit data through GUI provided in ESM.

- Security violation statistics and detailed report
- Login analysis report by user

Satisfying security function requirements: FAU_SAR.1, FAU_SAR.2, FAU_SAR.3

#### 6.1.5.5 Simple Attack Prevention (Audit.5)

The Simple attack prevention security function is to deny specific abuse actions detected as a violation conducted by a user over resources such as files and has the following detailed functions.

- Hidden process detect: Hacking programs such as 'Backdoor' will run without showing itself in process view of administrator. System admin can use this function to check whether there are any hidden processes.
- Prevent start program registration: Hacking programs will try to set itself in start program folder to be started automatically with Windows booting. This function can be used to prevent any unsecured programs' registration in the start program.
- Prevent registry start Key registration: Hacking programs will try to register itself in the registry's start-up Key (HKEY_LOCAL_MACHINE \ Software \ Microsoft \ Windows \ CurrentVersion \ Run) for automatic start with Windows booting. This function will prevent any unsecured programs' automatic start.

Satisfying security function requirement: FAU_SAA.3

### 6.1.6 Security Administration (Admin)

#### 6.1.6.1 Security Functions Management (Admin.1)

The TOE provides the following start-up and shutdown functions of security function handling parts for taking care of booting/shutdown of OS and exceptional works such as handling system maintenance job.

- Audit record collection and storage function
- Security kernel module function
- IP Filter as an interfacing function

The security officer can order start-up and shutdown of security function through GUI provided by RedCastle ESM and SecureOS. To do this job, the communication function of RedCastle SecureOS should be in use condition and this should be kept even if the security function is stopped.

The security officer only can perform the identification and authentication security functions before start-up the security function. The security function start-up can be done by the security officer through the following procedures. However, loading security module onto the operating system and unloading procedures will be done through separate automatic utility in the TOE.

- The security function start-up through ESM can be done only by the security officer who performs SecureOS identification and authentication.
- The security function start-up through GUI will be conducted with Administrator authority.
- SecureOS will start security function of log daemon first to collect and store the audit record.
- Then the reference monitor function will be started by replacing OS system calls to start access control security function.

The security function stop order by the security officer will be conducted according to the following procedures.

- The security function stop can be done by the security officer only who performs SecureOS identification and authentication process.

- SecureOS can stop the reference monitor function by recovering OS system calls to stop the access control security function.
- Then the security function of log daemon to collect and store the audit record will be stopped.

The security function's start-up and stop while the OS is booting and shutting down will be done by the service function of Windows operating system. To initiate the security function automatically when the OS is booted, the communication daemon of SecureOS will be initiated first. When the operating system is shutdown, the security function will be stopped first before the communication daemon is closed.

Satisfying security function requirements : FMT_MOF.1, FMT_SMF.1, FMT_SMR.1

#### 6.1.6.2 Hierarchical Category Management (Admin.2)

Security group (category) is corresponded to unclassified attributes among the sensitivity label of subject/object and it can be assigned by the security officer based on the organization's specific environment.

Security group in the TOE means subject or object scope defined in MAC policy and generally developed by reflecting the organization's system scope or departments.

The security officer can manage security group through GUI provided by RedCastle ESM.

- Security group query
- Security group add
- Security group delete
- Security group name change
- Security group transfer (Role change)

The security group's role which is representing security group's authority will be assigned as follows when a security group is added.

- Security officer group (SO): It will have default security group ID as 1(Security Admin) and it can't be added or deleted.
- System admin group (SA): It will have default security group ID as 2(System Admins) and a new system admin group should be assigned under the existing system admin group always. The newly added system admin group will be assigned the value between 3 – 127 for its security group ID in the order.
- Secured user group (MU): It doesn't have default security group ID and a new secured user group should be assigned under the security officer group or the existing secured user group. The newly added secured user group will be assigned the value between 3 – 127 for its security group iD in the order.

Satisfying Security Function Requirements : FMT_MSA.1(2), FMT_MSA.3(2), FMT_SMF.1, FMT_SMR.1

### 6.1.6.3 Labeled Users Management (Admin.3)

The secured user attributes in the TOE will be configured as follows.

- User identity
- Group identity user belongs
- Authentication data
- Sensitivity label
  - Security group
  - Security level
  - Security role : will be assigned based on security group role

In the secured user management function, sensitivity label can be assigned to the user and the security attributes for secured user can be queried, modified, and deleted. The user in the TOE will be classified as follows.

- Secured user : User doesn't have security level '0'. Secured user will be divided into security officer, system administrator, and secured user based on the assigned security group.
- System user : User who has security level of 0 and security group of 0.
- Administrator : User who classifies as super user in other operating system. The user will have security level and security group of 0.

The security officer can manage secured users' security attributes through the GUI provided by RedCastle ESM as the following rules.

- The OS Administrator can't have security attributes of secured user. The Administrator has security group of 0, security level of 0, and security role of 0 as default.
- User security attributes assignment (secured user registration): The security officer can assign security group and security level to the user belongs to the system account of OS. The security level of 7 is assigned as default.
  - Security officer: If a user has security group of 1, he will be registered as the security officer with security role of SO. If the user has security level of 1, he will have the authority to develop security policy. The security officer who has security level other than 1 doesn't have authority to develop security policy but he will have the authority to monitor security log by accessing SecureOS through ESM and query and search collected log.
  - System administrator: If a user is assigned as security group with the role of SA, he will be registered as system administrator with security role of SA.
  - Secured user : If a user is assigned as security group with the role of MU or other than security level of 0, he will be registered as secured user with security role of MU.
- User security attributes modification (security attributes modification): The security officer can modify security group and security level of a user who has the role of security officer, system administrator, and secured user. The security officer can modify authentication failure threshold numbers or revoke secured user's authentication prevention through the GUI provided by ESM.

- User security attributes revocation (secured user deletion): The security officer can revoke all security attributes of a user who is assigned as role of security officer, system administrator, and secured user and delete the user from the secured user list.
- User security attributes query: The security officer can query secured user security attributes including all user information in the system through the GUI provided by ESM.

Satisfying security function requirements : FIA_ATD.1, FMT_MSA.1(2), FMT_MSA.3(2), FMT_MTD.1(2), FMT_REV.1(1), FMT_SMF.1, FMT_SMR.1

#### 6.1.6.4 Labeled Objects Management (Admin.4)

In the TOE, a file(object) security attributes will be configures as follows.

- File and directory name
- Sensitivity label
    - Security group
    - Security level
    - Security role: will be selected based on security group role

In the TOE, the subject security attributes will be assigned to a newly created file automatically. However, this function will be used when the security officer tries to assign the security attributes specially.

In the object security attributes management function, the sensitivity label can be assigned to a file(object) and the security attributes of a file can be queried, modified, and deleted. The security officer can manage the object security attributes with the following rules through the GUI provided by RedCastle ESM.

- The security attributes of a file will have a default value (UX, 0, 0). The file created by the user with system user(UX) authority will have the same security attributes too.

- User security attributes modification (security attributes modification): The security officer can modify security group and security level of a file. The security level of 7 will be assigned as default.
  - Security officer file: If security group of 1 is assigned to a file, the file will be security officer file with security role of SO.
  - System administrator file: If security group with role of SA is assigned, the file will have security role of SA.
  - Secured user file: If security group with role of MU is assigned, the file will have security role of MU.
- File security attributes revocation (security attributes delete): The security officer can revoke all security attributes of a file.
- File security attributes query: The security officer can query security attributes information of a file through the GUI provided by ESM.

Satisfying security function requirements: FMT_MSA.1(2), FMT_MSA.3(2), FMT_REV.1(2), FMT_SMF.1, FMT_SMR.1

#### 6.1.6.5 Labeled Processes Management (Admin.5)

In the TOE, a process (subject) security attributes will be configured as follows.

- Process ID
- Owner Identity
- Sensitivity label
  - Security group
  - Security level
  - Security role status : will be selected based on security group role and owner identity

The TOE is designed to assign security attributes of subject to the newly created process automatically and the security officer can use this function to query the security attributes of a process (subject).

The security officer can query subject security attributes through the GUI provided by RedCastle ESM.

Satisfying security function requirements: FMT_MSA.1(2), FMT_SMF.1, FMT_SMR.1

### 6.1.6.6 ACL Policy Management (Admin.6)

In the TOE, the security officer (SO) is able to configure following rule of discretionary access control by file or file group for ACL policies management.

- Subject information : Default value - Any
  - The owner of subject
  - The security category of subject
  - The security role status of subject
- Subject program name : Default value - NULL
- Operation : Default value – Allow access
  - read
  - write
  - execute
  - create
  - delete
  - rename

SO manages ACL policies through GUI provided in RedCastle ESM.

- ACL group management : query, add, delete, rename
- Individual ACL policy management : query, add, modify, delete

- Group file ACL policy management : query, add, modify, delete
- Registry ACL policy management : query, add, modify, delete

Security Requirements to be satisfied: FMT_MSA.1(1), FMT_MSA.3(1), FMT_SMF.1, FMT_SMR.1

#### 6.1.6.7 Allow/Deny List Management (Admin.7)

The security officer(SO) is able to add, search, and delete the following rules by GUI(Graphical User Interface) and CLI(Command Line Interface) for managing the policy that will allow or deny explicitly the access of subjects to objects based on the security attribute.

- Command Control List
- Kill Control List
- Sharing Directory Access Allow List
- Execution Bypass Allow List

In the TOE, the security officer can configure the allow/deny list as follows.

- The command control list keeps the commands (file name not included the path) list which can be executed by the security officer (SO) only and the security officer can query the command list and add or delete the new commands.
- The Kill control list keeps the process (executed process command) list which can be executed by the security officer (SO) only and the security officer can query, add, and delete the processes. The security officer can configure the option for the corresponding process to allow system administrator to kill it.
- The sharing directory access allow list keeps Windows sharing directory list allowing its access for user and the security officer can query, add, and delete it. The unregistered user’s access to the unregistered sharing directory in the list will not be allowed.

- The Execution Bypass Allow List keeps program names bypass level-based MAC regardless of subject security attributes and reconfigurable security attributes. When the enlisted program is executed, the execution will be allowed bypassing level-based MAC regardless of subject security attributes and the allowed subject program will be reassigned security attributes prescribed in the list.

Satisfying Security Function Requirements : FMT_MSA.1(1), FMT_MSA.3(1), FMT_SMF.1, FMT_SMR.1

#### 6.1.6.8 Audit Configuration (Admin.8)

A security officer(SO) in the TOE is provided functions that are able to configure path, file, and alarm of audit storage.

- Audit storage file location
- Audit storage size: file size, file numbers
- Countermeasure setting for audit storage saturation
- Potential violation analysis threshold setting : time interval, accumulated violation numbers
- Countermeasure setting for detected potential violation analysis
- Setting administrator mail address to notify

Satisfying security function requirements : FMT_MTD.1(1), FMT_SMF.1, FMT_SMR.1

#### 6.1.6.9 ESM User Management (Admin.9)

By using the GUI(Graphical User Interface) provided in RedCastle ESM, it is possible to add/delete ESM administrator and user, and to change own ESM password.

The first registered user into an ESM when it is initiated after its installation will have the authority to being an ESM administrator. And the ESM administrator can register and delete new ESM user afterward. Required information for ESM user registration is as follows.

- ESM user ID
- ESM user password

An authentication data will be generated combining ID with password and this will be encrypted using SEED and SHA-1 algorithm.

The password to be used for an ESM user registration or its changes must satisfy the following conditions.

- Password length: between 8 and 15 characters
- Acceptable password characters.
  - 52 alphabetic letters (lowercase or uppercase)
  - 10 numeric digits (0-9)
  - 16 special characters (!,@,#,$,%,^,&,*,(,),+,=,<,>,:,;)
- A password must contain at least one character of alphabetic, numeric, and special characters.
- Allow to use the consecutive alphabetic or numeric
- Allow to use the repetition of characters

Security Requirements to be satisfied: FIA_SOS.1, FMT_MTD.1(2), FMT_MTD.1(3), FMT_REV.1(1), FMT_SMF.1, FMT_SMR.1

#### 6.1.6.10 Security Password Management (Admin.10)

This TOE provides function that can register and change authentication data, namely security password.

A registration and a change of security password are possible only related user. A security officer (SO) is possible to change a security password by GUI and CLI. A system administrator except SO and a multi-label security user (MU) are possible to change a security password of self.

If SO registers SA (system administrator) as MU in the labeled users management(Admin.3), a password of that user is initialized.

An authentication data is generated mixing ID and password and this is encrypted using SEED algorithm and SHA-1 algorithm.

When security password registered or changed, a password must satisfy following conditions.

- Password length: between 8 and 15 characters
- Acceptable password characters.
  - 52 alphabetic letters (lowercase or uppercase)
  - 10 numeric digits (0-9)
  - 16 special characters (!,@,#,$,%,^,&,*,(,),+,=,<,>,:,;)
- A password must contain at least one character of alphabetic, numeric, and special characters.
- Allow to use the consecutive alphabetic or numeric
- Allow to use the repetition of characters

Security Requirements to be satisfied: FIA_SOS.1, FMT_MTD.1(3), FMT_SMF.1, FMT_SMR.1

### 6.1.6.11 Security Functions Configuration (Admin.11)

After the security function activated, the TOE provides the following functions to the security Officer to configure the security operating environment by GUI.

- Multi-level based MAC : On, Warning
- ACL based DAC : On, Warning, Off
- Allowed/Denied list based DAC : On, Warning, Off
  - Command execution control
  - Process kill prevention list
  - Sharing directory access allow list apply

- Simple attack prevention : On, Warning, Off
  - Detecting hidden process
  - Prevent start program registration
  - Prevent registry start-up key registration
- Security module hiding : On, Off
- Restriction of process attribute monitoring : On, Off

Security Requirements to be satisfied: FMT_MTD.1(4), FMT_SMF.1, FMT_SMR.1

#### 6.1.6.12 System Services Management (Admin.12)

The TOE provides GUI to configure and manage policies for specific functions exist outside of the TOE. The followings are functions providing management GUI connecting with the TOE.

- IP Filter function
  - Inbound Network connecting control function
  - Outbound Network connecting control function
- System Monitoring function
  - System performance (CPU capacity, Memory occupation) monitoring function
  - Process operation monitoring function
  - Process CPU occupation restrict function
  - Disk usage monitoring function
- System account management function
  - System account information add/modify/delete function
  - System account password policy setting function
  - System account password modify function
  - System group add/modify/delete function

Satisfying security function requirements : FMT_SMF.1, FMT_SMR.1

### 6.1.7 TSF Protection (Protect)

#### 6.1.7.1 Abstract machine testing (Protect.1)

The TOE provides abstract machine and TSF operation testing function to check operating system state and operation state of the TOE on the system.

- System status query: CPU occupation, Memory occupation, booting elapse time
- SecureOS status query : ESM connection time, security module version, security module operation status, log daemon operation status, IP Filter operation status
- Security log real-time query

Security Requirements to be satisfied: FMT_SMF.1, FPT_AMT.1, FPT_TST.1

#### 6.1.7.2 Integrity Checking Functions (Protect.2)

For TSF's secure operation, the TOE provides integrity checking functions over TSF's execute file, TSF's data file, and files which administrator select.

- Integrity checking target file registration
- File deletion from the integrity checking target list
- Integrity check execution
- Integrity check results review and update

Execution files of RedCastle ESM and RedCastle SecureOS is registered as a default file for integrity checking target and will not be deleted from the list.

In the case of a first integrity checking, the integrity checking value will be stored and afterward created integrity value and stored integrity checking value will be compared. The TOE will use SHA-2 for integrity checking.

Security Requirements to be satisfied: FMT_MTD.1(4), FMT_SMF.1, FPT_TST.1

#### 6.1.7.3 ESM Screen Saving (Protect.3)

The TSF can lock the interactive sessions of security officer(SO) while he is inactive and unlock the sessions through the identification and authentication of the administrator. The TOE provides a session locking of inactivation through the ESM Screen Saving function.

The screen saving function is applied to Windows system that RedCastle ESM is installed and the queuing time will be set by using GUI(Graphical User Interface). If the ESM user's inactive state lasts more than the queuing time set in advance, then the Screen Saving function will start. The screen saving function can be unlocked only by the authentication of ESM user's password in the TOE.

Security Requirements to be satisfied: FMT_MTD.1(4), FMT_SMF.1, FTA_SSL.1

#### 6.1.7.4 Secure Communication (Protect.4)

The TOE installed the communication Server in the RedCastle SecureOS and the communication Client in the RedCastle ESM.

The communication between RedCastle SecureOS and RedCastle ESM uses the SSL(Secure Socket Layer openssl-0.9.8e) version 3 protocol. For key exchange and authentication method, anonymous authentication mechanism (anonymous DH) is used. AES encryption algorithm (256 bit cipher key) and SHA Hash algorithm will be selected for cipher-suite (ADH-AES256-SHA).The SSL protocol encodes data and therefore authentication information is secured by SSL protocol.

The secure communication server provides access control function to the RedCastle ESM by identifying whether it is connectable by using SO's account and ESM IP address for the Security Officer to manage RedCastle SecureOS.

Security Requirements to be satisfied: FIA_UAU.4, FPT_ITT.1, FTP_ITC.1

## 6.2 Assurance Measure

The ST assurance measure consists of assurance components in CC Part 3 and its assurance level is EAL3+.

**[Table 6-3] Mapping Assurance Requirements to Measures**

| Assurance Component | | Measures |
|---|---|---|
| ACM_CAP.3 | Authorization controls | Configuration management |
| ACM_SCP.1 | TOE CM coverage | |
| ADO_DEL.1 | Delivery procedures | Delivery procedures |
| ADO_IGS.1 | Installation, generation, and start-up procedures | Installation guidance |
| ADV_FSP.1 | Informal functional specification | Functional specification |
| ADV_HLD.2 | Security enforcing high-level design | High Level Design |
| ADV_IMP.2 | Implementation of the TSF | Implementation representation |
| ADV_LLD.1 | Descriptive low-level design | Low Level Design |
| ADV_RCR.1 | Informal correspondence demonstration | Correspondence information of the functional specification |
| AGD_ADM.1 | Administrator guidance | Administrator guidance |
| AGD_USR.1 | User guidance | User guidance |
| ALC_DVS.1 | Identification of security measures | Life cycle support |
| ALC_TAT.1 | Well-defined development tools | |
| ATE_COV.2 | Analysis of coverage | Testing |
| ATE_DPR.2 | Testing: low-level design | |
| ATE_FUN.1 | Functional testing | |
| ATE_IND.2 | Independent testing-sample | N/A(evaluator) |
| AVA_MSU.1 | Examination of guidance | N/A(evaluator) |
| AVA_SOF.1 | Strength of TOE security function evaluation | Vulnerability analysis |

| | | |
|---|---|---|
| AVA_VLA.2 | Independent vulnerability analysis | |

## 6.2.1 Configuration Management

### 6.2.1.1 CM Documents

CM document is a means or way to verify that functional requirements and specifications can be realized through the TOE implementation. Configuration management satisfies this objective by applying and controlling the rules in the process to redefine and modify the TOE and related information. Configuration management system assures its controlling integrity of TOE by providing all measures to trace modifications and verifies all modifications’ approval. In the TOE, configuration tool ‘Clear Case’ is used.

The TOE’s CM document handles ‘ACM_CAP, CM Capabilities’ and ‘ACM_SCP, CM Scope’ family of ‘ACM, Configuration management’ class. In ‘ACM_CAP” family, ‘ACM_CAP.3’ component will be handled and in ‘(ACM_SCP)’ family, ACM_SCP.1’ will be handled.

Satisfying assurance requirements: ACM_CAP.3, ACM_SCP.1

## 6.2.2 Delivery and Operation

### 6.2.2.1 Delivery Procedure Documents

In the Delivery Procedure Document, requirements for accurate delivery of the TOE will be defined. In the TOE delivery document, ‘ADO_DEL, Delivery’ component of ‘ADO_DEL, Delivery family in ‘ADO, Delivery and operation’ class will be handled.

Satisfying assurance requirements: ADO_DEL.1

#### 6.2.2.2 Installation Guide

The installation guide will define the requirements for installation, generation, and initiation of the TOE. The installation guide of TOE will cover 'Installation, Generation, and Start-up procedure (ADO_IGS.1) component of '(ADO_IGS, Installation, generation and start-up)' family in '(ADO, Delivery and operation)' class.

Satisfying assurance requirements: ADO_IGS.1

### 6.2.3 Development

#### 6.2.3.1 Functional Specification

The Functional Specification is basic description of interface and operation that TSF user can see and showing security functional requirements of the TOE. The Functional Specification verifies all security functional requirements are covered.

The Functional Specification of the TOE covers 'ADV_FSP.1, Informal Functional Specification' component described in 'ADV_FSP, Functional specification' family of 'ADV, Development' class.

Also, it provides conformity between the TOE summarized specification and functional specification based on 'ADV_RCR.1, Informal Conformity Verification' described in 'ADV_RCR, Representation correspondence' family.

Satisfying assurance requirements: ADV_FSP.1, ADV_RCR.1

#### 6.2.3.2 High Level Design

The High-level Design will describe the TSF with main components (subsystems) and explain the relationship between the main components and its providing functions. The requirements for the High-level Design are to assure of providing proper structure for implementing TOE security functional requirements.

The High-level Design of the TOE covers 'ADV_HLD.2' described in 'ADV_HLD, High-level design' family of 'ADV, Development' class.

Also, it provides the conformity between the functional specification and high-level design based on ‘ADV_RCR.1, Informal Conformity Verification’ component described in ‘ADV_RCR, Representation correspondence’ family.

Satisfying assurance requirements: ADV_HLD.2, ADV_RCR.1

#### 6.2.3.3 Low Level Design

The Low Level Design describes the TSF internal operation with co-relationship and dependency of the modules. The Low Level Design provides the assurance that the TSF subsystem is specified accurately and efficiently. The Low Level Design describes the objective, function, interface, dependency, implementation of TSP-performing function for each module of the TSF.

The Low Level Design of the TOE covers ‘ADV_RCR.1’ component described in ‘ADV_LLD, Low-level design’ family of ‘ADV, Development’ class.

Also, it provides the conformity between High-level Design and Low-level Design based on ‘ADV_RCR.1, Informal Conformity Verification’ described in ‘ADV_RCR, Representation correspondence’ family.

Satisfying assurance requirements: ADV_LLD.1, ADV_RCR.1

#### 6.2.3.4 Implementation Verifying Specification

The Implementation Verifying Specification will be represented in source code format and supports the analysis by confirming specified internal operations of the TSF.

The Implementation Verifying Specification covers ‘ADV_IMP.2’ component described in ‘ADV_IMP, Implementation representation’ family of ‘ADV, Development’ class.

Also, it provides the conformity between the Low Level Design and Implementation Verifying Specification based on 'ADV_RCR.1' component described in 'ADV_RCR, Representation correspondence' family.

Satisfying assurance requirements: ADV_IMP.2, ADV_RCR.1

#### 6.2.3.5 Representation Correspondence

The Representation Correspondence of the TSF covers how the most specified TSF representations can realize the requirements correctly and completely.

The Representation Correspondence of the TOE will be included in the TOE summarized specification, functional specification, high-level design, low-level design, and implementation verifying specification. The Representation Correspondence provides the conformity between the low-level design and verification specification based on 'ADV_RCR.1' component described in 'ADV_RCR, Representation correspondence' family.

Satisfying assurance requirements: ADV_RCR.1

### 6.2.4 Guidance

#### 6.2.4.1 Admin Guidance

The Admin Guidance is documented material can be used by people responsible for configuration, maintenance, and management of the TOE in accurate method to maximize security. Since the safe operation of the TOE is depending on accurate conduction of the TSF, the people responsible for conducting these functions are trusted by the TSF. The Admin Guide is helping the security officer to understand security functions provided by the TOE including the function required to perform by the security officer and functions providing important security information.

The admin Guidance of the TOE covers 'AGD_ADM.1' component of 'AGD_ADM, Administrator guidance' family in 'AGD, Guidance documents' class.

Also, the Admin Guidance includes misuse analysis and covers ‘AVA_MSU.1’ component of ‘AVA_MSU, Misuse’ family in ‘AVA, Vulnerability assessment’ class.

Satisfying assurance requirements: AGD_ADM.1, AVA_MSU.1

#### 6.2.4.2 User Guidance

The User Guidance means a material can be used by TOE user and people utilize the external interface of the TOE other than the security officer. The User Guidance describes security functions provided by the TSF and provides directions including alert and guides.

The User Guidance of TOE covers ‘AGD_USR.1’ component of ‘AGD_USR, User guidance’ family in ‘AGD, Guidance documents’ class.

Also, the User Guidance includes the misuse analysis and covers ‘AVA_MSU.1’ component of ‘AVA_MSU, Misuse’ family in ‘AVA, Vulnerability assessment’ class.

Satisfying assurance requirements: AGD_USR.1, AVA_MSU.1

### 6.2.5 Life Cycle Support

#### 6.2.5.1 Life cycle support document

The Life cycle support founds rules and controls in the specifying process of TOE during its development and maintenance. When the security analysis and evidence creation is performed regularly as a part of development process and operation support activity, the reliability for conformity between the TOE and its security requirements will be increased.

The Life Cycle Support Document of the TOE covers ‘ALC_DVS, Development security’ and ‘ALC_TAT, Tools and techniques’ family of ‘ALC, Life cycle support’ class.

Also, in ‘ALC_DVS’ family, ‘ALC_DVS.1’ component is covered and in ‘ALC_TAT’ family, ‘ALC_TAT.1’ component will be covered.

Satisfying assurance requirements : ALC_DVS.1, ALC_TAT.1

## 6.2.6 Tests

### 6.2.6.1 Tests document

The Tests help to verify satisfaction of TOE security function requirements. The Tests can't configure the TOE to perform specified parts only but it can assure the TOE satisfies security function requirements at least.

The tests of TOE covers 'ATE_COV, Coverage', 'ATE_DPT, Depth', 'ATE_FUN, Functional testing', and 'ATE_IND, Independent testing' families of 'ATE, Tests' class. The 'ATE_IND, Independent Test' family will not be covered here since it is the test should be done by the evaluator.

The 'ATE_COV' family will cover 'ATE_COV.2, the analysis of test scope' component, 'ATE_DPT' family will cover 'ATE_DPT.2' component, and 'ATE_FUN' family will cover 'ATE_FUN.1' component.

Satisfying assurance requirements : ATE_COV.2, ATE_DPR.2, ATE_FUNC.1, ATE_INFD.2(N/A)

## 6.2.7 Vulnerability Assessment

### 6.2.7.1 Vulnerability Assessment Document

The Vulnerability Analysis is to determine whether a user violates the TSF by using identified vulnerabilities while the TOE configuration and expected operation were evaluating or through other means (for example Fault Assumption). Also, the TOE security function strength will be assumed by its clarification and it shows the capability of related security functions how to response for the identified threats.

The Vulnerability Assessment Document of TOE covers 'AVA_SOF, Strength of TOE security functions' and 'AVA_VLA, Vulnerability analysis' families of 'AVA, Vulnerability assessment' class.

The ‘TOE Security Function Strength, (AVA_SOF)’ family covers ‘Evaluation of TOE Security Function Strength, (AVA_SOF.1)’ component and ‘Vulnerability Analysis, (AVA_VLA)’ family covers ‘Independent Vulnerability Analysis, (AVA_VLA.2)’ component.

Satisfying assurance requirement : AVA_SOF.1, AVA_VLA.2

#### 6.2.7.2 Misuse Analysis

The Misuse Analysis inspects whether the security officer and user of the TOE trust the safety of TOE with proper reasons even though the TOE is configured or used in unsafe way. The Misuse Analysis can be provided as included in the existed user or admin guidance or separated.

The Misuse Analysis of TOE will be included in the existed user or admin guidance and covers ‘AVA_MSU.1’ component of ‘AVA_MSU, Misuse’ family in ‘AVA, Vulnerability assessment’ class.

Satisfying assurance requirement : AVA_MSU.1

# 7 PP Claims

## 7.1 PP reference

This Security Target claims conformance with the “Label-based Access Control System Protection Profile for Government v1.1, 2006-05-17 (LACSPP)”.

## 7.2 PP Refinements and Additions

### 7.2.1 Refinement of PP functional requirements

The security functional requirements of this ST are the same as the ones of the Protection Profile.

**[Table 7-1] PP Refinements and Additions**

| Class | PP Component | | Refinement |
|---|---|---|---|
| Security Audit | Security alarms | FAU_ARP.1.1 | Refinement |
| | Audit data generation | FAU_GEN.1.1<br>FAU_GEN.1.2 | |
| | User identity association | FAU_GEN.2.1 | |
| | Potential violation analysis | FAU_SAA.1.1<br>FAU_SAA.1.2 | |
| | Simple attack heuristics | FAU.SAA.3.1<br>FAU.SAA.3.2<br>FAU.SAA.3.3 | Refinement<br>Refinement |
| | Audit review | FAU_SAR.1.1<br>FAU_SAR.1.2 | |
| | Restricted audit review | FAU_SAR.2.1 | |
| | Selectable audit review | FAU_SAR.3.1 | Refinement |
| | Selective audit | FAU_SEL.1.1 | Refinement |
| | Protected audit trail storage | FAU_STG.1.1<br>FAU_STG.1.2 | |
| | Action in case of possible audit | FAU_STG.3.1 | Refinement |

| | | | |
|---|---|---|---|
| | data loss | | |
| | Prevention of audit data loss | FAU_STG.4.1 | Refinement |
| User Data Protection | Subset access control | FDP_ACC.1.1 | Assignment, Refinement |
| | Security attribute based access control | FDP_ACF.1.1<br>FDP_ACF.1.2<br>FDP_ACF.1.3<br>FDP_ACF.1.4 | Assignment, Refinement<br>Assignment, Refinement<br>Assignment, Refinement<br>Assignment, Refinement |
| | Subset information flow control | FDP_IFC.1.1 | Assignment, Refinement |
| | Hierarchical security attributes | FDP_IFF.2.1<br>FDP_IFF.2.2<br>FDP_IFF.2.3<br>FDP_IFF.2.4<br>FDP_IFF.2.5<br>FDP_IFF.2.6<br>FDP_IFF.2.7 | Selection<br>Assignment, Refinement<br>Assignment, Refinement<br>Assignment, Refinement<br>Assignment, Refinement |
| | Import of user data without security attributes | FDP_ITC.1.1<br>FDP_ITC.1.2<br>FDP_ITC.1.3 | Refinement |
| | Subset residual information protection | FDP_RIP.1.1 | |
| Identification & Authentication | Authentication failure handling | FIA_AFL.1.1<br>FIA_AFL.1.2 | Assignment, Refinement |
| | User attribute definition | FIA_ATD.1.1 | Refinement |
| | Verification of secrets | FIA_SOS.1.1 | Assignment, |

| | | | |
|---|---|---|---|
| | | | Refinement |
| | Timing of authentication | FIA_UAU.1.1<br>FIA_UAU.1.2 | |
| | Single-use authentication mechanisms | FIA_UAU.4.1 | Assignment, Refinement |
| | Protected authentication feedback | FIA_UAU.7.1 | Assignment, Refinement |
| | User identification before any action | FIA_UID.2.1 | |
| | User-subject binding | FIA_USB.1.1<br>FIA_USB.1.2<br>FIA_USB.1.3 | |
| Security Management | Management of security function behavior | FMT_MOF.1.1 | Assignment, Refinement Selection |
| | Management of security attributes (DAC) | FMT_MSA.1(1).1 | |
| | Management of security attributes (MAC) | FMT_MSA.1(2).1 | |
| | Static attribute initialization (DAC) | FMT_MSA.3(1).1<br>FMT_MSA.3(1).2 | |
| | Static attribute initialization (MAC) | FMT_MSA.3(2).1<br>FMT_MSA.3(2).2 | |
| | Management of TSF data (Audit data) | FMT_MTD.1(1).1 | |
| | Management of TSF data (Identification and authentication data) | FMT_MTD.1(2).1 | |
| | Management of TSF data (authentication data) | FMT_MTD.1(3).1 | |
| | Management of TSF data (TSF data) | FMT_MTD.1(4).1 | |
| | Revocation (User) | FMT_REV.1(1).1<br>FMT_REV.1(1).2 | |
| | Revocation (Object) | FMT_REV.1(2).1 | |

| | | | |
|---|---|---|---|
| | | FMT_REV.1(2).2 | |
| | Specification of management functions | FMT_SMF.1.1 | Assignment |
| | Security roles | FMT_SMR.1.1<br>FMT_SMR.1.2 | Refinement |
| TSF Protection | Abstract machine testing | FPT_AMT.1.1 | |
| | Basic internal TSF data transfer protection | FPT_ITT.1.1 | Selection |
| | Non-bypassability of the TSP | FPT_RVM.1.1 | |
| | TSF domain separation | FPT_SEP.1.1<br>FPT_SEP.1.2 | |
| | Reliable time stamps | FPT_STM.1.1 | |
| | TSF testing | FPT_TST.1.1<br>FPT_TST.1.2<br>FPT_TST.1.3 | |
| TOE Access | TSF-initiated session locking | FTA_SSL.1.1<br>FTA_SSL.1.2 | Assignment, Refinement<br>Assignment, Refinement |
| Trusted Path/Channels | Inter-TSF trusted channel | FTP_ITC.1.1<br>FTP_ITC.1.2<br>FTP_ITC.1.3 | |

### 7.2.2 PP Security Function Requirements Addition

The additional security function requirements in the ST are as follows.

**[Table 7-2] Additional assurance requirements**

| COMPONENT | DESCRIPTION |
|---|---|
| FPT_ITT.1 (Protection of Internal TSF Data) | Provides trusted communication channel to prevent TSF data exposure from communication between separated TOE parts |
| FAU_SAA.3 | TOE provides capability to indicate signature event occurrence |

| | |
|---|---|
| (Simple Attack Intelligence) | which can be a major threat for security policy implementation by using audit target events (system call). |

### 7.2.3 PP Assurance Requirements Addition

There are no additional security assurance requirements and the requirements of this ST are the same as the ones of Protection Profile.

# 8 Rationale

This section presents the evidence used in the Security Target evaluation. This evidence supports the claims that the ST is a complete and cohesive set of requirements, that a conformant TOE would provide an effective set if IT security countermeasures within the security environment, and that the TOE summary specification addresses the requirements. The rationale also demonstrates that any PP conformance claims are valid.

## 8.1 Security Objectives Rationale

Security objectives rationale will prove that the specific security objectives are appropriate, sufficient to handle security incidents, and indispensable.

Security objectives rationale will verify the followings.

- Each assumption, threat, and organizational policy will be cared by at least one of security objectives.
- Each security objective will at least cares one of assumptions, threats, and organizational policies.

### 8.1.1 Rationale of Security Objectives for TOE

**[Table 8-1] Mapping Objectives to assumptions, threats, polices**

| SECURITY OBJECTIVES<br>SECUIRTY ENVIRONMENT | TOE SECURITY OBJECTIVES | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | O.AUDIT | O.MAC | O.CODE | O.MANAGE | O.INTEGRITY | O.LABEL | O.IA | O.DAC | O.PROTECT | O.RESIDUE |
| T.CODE | | | X | | | | | | | |
| T.AUDIT | X | | | | | | | | | |
| T.INTEGRITY | | | | | X | | X | | X | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| T.UAUTH | | | | | | | X | | | |
| T.BYPASS | | | | | | | X | | X | |
| T.RESIDUE | | | | | | | | | | X |
| P.AUDIT | X | | | | | | | | | |
| P.MAC | | X | | | | X | | | | |
| P.LABEL | | X | | | | X | | | | |
| P.IA | | | | | | | X | | | |
| P.MANAGE | | | | X | | | | | | |
| P.CIPHER | | | | X | | | | | | |
| P.DAC | | | | | | | | X | | |

**O.AUDIT**

As the security objective assures the TOE to provide a measure to record and review security associated events, it is necessary to correspond threat **T.FAIL TO RECORD** and support organizational security policy **P.AUDIT**.

**O.MAC**

As the security objective assures the TOE to control information flow based on information and user's security level, it is necessary to correspond security policy **P.MAC** and **P.LABEL**.

**O.CODE**

As the security objective required to check whether the code developed by developer has defects and inspect whether the defected code affects internal components of the TOE, it is necessary to correspond the assumption **T.CODE**.

**O.MANAGE**

As the security objective provides security officer with the measure to access the TOE securely and manage, it is supporting **P.MANAGE** and associating with **P.CIPHER** when the TOE is managed by remote.

**O.INTEGRITY**

Since the TOE must protect the TSF data or the reliable data from the unauthorized disclosure, modification, and deletion, **O.INTEGRITY** is required to counter **T.INTEGRITY**.

**O.LABEL**

Since the TOE must assign and revoke the security label of the subject and the object according to the organization access control policies, **O.LABEL** is required to counter **P.LABEL** and **P.MAC**.

**O.IA**

Since the TOE must identify a user uniquely and ensure that only authorized users gain access to the TOE and its resources, O.IA is required to counter **T.INTEGIRTY**, **T.UAUTH**, **T.BYPASS** and **P.IA.**

**O.DAC**

Since the TOE must enforce the access to resources on the basis of the identity of user or group, **O.DAC** is required to counter **P.DAC.**

**O.PROTECT**

Since the TOE must protect itself from the deactivation or the modification of the TOE, O.PROTECT is required to counter **T.INTEGRITY and T.BYPASS**.

**O.RESIDUE**

Since the TOE must ensure that any information contained in a protected resource is not released when the resource is recycled, O.RESIDUE is required to counter **T.RESIDUE**.

### 8.1.2 Rationale of Security Objectives for Environment

**[Table 8-2] Mapping Objectives for Environment**

| OBJECTIVES<br>SECURITY ENVIRONMENT | | SO for TOE Environment | | | | ADDED OBJECTIVES | |
|---|---|---|---|---|---|---|---|
| | | OE.LOCATE | OE.ADMINS | OE.MANAGE | OE.PATCH | OE.SSL | OE.TIME |
| A.LOCATE | | X | | | | | |
| A.ADMINS | | | X | | | | |
| A.PATCH | | | | | X | | |
| TE.MISUSE | | | X | X | | | |
| TE.INSTALL | | | X | X | | | |
| P.ADMINS | | | | X | | | |
| ADDED ASSUMPTIONS | A.SSL PROTOCOL | | | | | X | |
| | A.TIME | | | | | | X |

**OE.LOCATE**

Since the TOE shall be located within controlled access facilities and protected from unauthorized physical modification, **OE.LOCATE** is countered to **A.LOCATE**.

**OE.ADMINS**

Since it is ensured that one or more competent individuals who are assigned to manage the TOE are assumed not to be careless, willfully negligent or hostile, OE.ADMINS is countered to **TE.MISUSE, TE.INSTALL,** and **A.ADMINS**.

**OE.MANAGE**

Since the TOE must be installed securely, configured, managed, and used by an only authorized administrator, OE.MANAGE is countered to **TE.MISUSE**, **TE.INSTALL,** and **P.ADMIN**.

**OE.PATCH**

Since the operating system will be patched the vulnerabilities and removed the useless services before the installing the TOE, **OE.PATCH** is countered to **A.PATCH**.

**OE.SSL PROTOCOL**

Since the TOE uses the SSL(openssl-0.9.8e) version 3 protocol for ensuring the secure communication which is provided by IT environment, **OE.SSL PROTOCOL** is countered to **A.SSL PROTOCOL**.

**OE.TIME**

Since the TOE uses the reliable timestamp which is provided by IT environment, **OE.TIME** is countered to **A.TIME**.

## 8.2 Security Requirements Rationale

The security requirements rationale shall demonstrate that the set of security requirements (TOE and environment) is suitable to meet and traceable to the security objectives.

### 8.2.1 Rationale for TOE Security Requirements

The rationale of TOE Security Requirements will verify the followings.

- Each TOE security objective will be cared by at least one of TOE security function requirements. However, **O.CODE** will be cared by assurance requirements.

- Each TOE security requirement cares at least one of TOE security objectives.

**[Table 8-3] Mapping security requirements to objectives**

| ST<br>Security Functional Requirements | O.AUDIT | O.MAC | O.MANAGE | O.INTEGRITY | O.LABEL | O.IA | O.DAC | O.PROTECT | O.RESIDUE |
|---|---|---|---|---|---|---|---|---|---|
| FAU_ARP.1 | X | | | X | | | | X | |
| FAU_GEN.1 | X | | | | | | | | |
| FAU_GEN.2 | X | | | | | | | | |
| FAU_SAA.1 | X | | | | | | | | |
| FAU_SAA.3 | X | | | | | | | | |
| FAU_SAR.1 | X | | | | | | | | |
| FAU_SAR.2 | X | | | | | | | | |
| FAU_SAR.3 | X | | | | | | | | |
| FAU_SEL.1 | X | | | | | | | | |
| FAU_STG.1 | X | | | | | | | | |
| FAU_STG.3 | X | | | | | | | | |
| FAU_STG.4 | X | | | | | | | | |
| FDP_ACC.1 | | | | | | | X | | |
| FDP_ACF.1 | | | | | | | X | | |
| FDP_IFC.1 | | X | | | | | | | |
| FDP_IFF.2 | | X | | | | | | | |
| FDP_ITC.1 | | X | | | | | | | |
| FDP_RIP.1 | | | | | | | | | X |
| FIA_AFL.1 | | | | | | X | | | |
| FIA_ATD.1 | | | | | | X | | | |
| FIA_SOS.1 | | | | | | X | | | |
| FIA_UAU.1 | | | X | X | | X | | | |
| FIA_UAU.4 | | | | | | X | | | |
| FIA_UAU.7 | | | | | | X | | | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| FIA_UID.2 | | | X | X | | X | | | |
| FIA_USB.1 | | | | | | X | | | |
| FMT_MOF.1 | | | X | | | | | | |
| FMT_MSA.1(1) | | | X | | | | X | | |
| FMT_MSA.1(2) | | X | X | | X | | | | |
| FMT_MSA.3(1) | | | X | | | | X | | |
| FMT_MSA.3(2) | | X | X | | X | | | | |
| FMT_MTD.1(1) | | | X | | | | | | |
| FMT_MTD.1(2) | | | X | | | | | | |
| FMT_MTD.1(3) | | | X | | | | | | |
| FMT_MTD.1(4) | | | X | | | | | | |
| FMT_REV.1(1) | | | X | | X | | | | |
| FMT_REV.1(2) | | | X | | X | | | | |
| FMT_SMF.1 | | | X | | | | | | |
| FMT_SMR.1 | | | X | | | | | | |
| FPT_AMT.1 | | | | X | | | | X | |
| FPT_ITT.1 | | | X | | | | | | |
| FPT_RVM.1 | | | | | | | | X | |
| FPT_SEP.1 | | | | | | | | X | |
| FPT_STM.1 | X | | | | | | | | |
| FPT_TST.1 | | | | X | | | | X | |
| FTA_SSL.1 | | | | X | | | | X | |
| FTP_ITC.1 | | | X | | | | | | |

**FAU_ARP.1 Security alarm**

This component satisfies **O.AUDIT** because the TSF ensures the functions taking the list of the least disruptive actions upon detection of a potential security violation.

**FAU_GEN.1 Audit data generation**

This component satisfies **O.AUDIT** because the TSF ensures the functions generating an audit record for the auditable events.

### FAU_GEN.2 User identity association

This component satisfies **O.AUDIT** because the TSF ensures the functions associating 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 the TSF ensures the functions enforcing the rules for monitoring audited events and indicating a potential violation of the TSP.

### FAU_SAA.3 Simple attack heuristics

This component satisfies **O.AUDIT** because the TSF ensures the functions detecting the signature events from the auditable events and indicating a potential violation.

### FAU_SAR.1 Audit review

This component satisfies **O.AUDIT** because the TSF ensures the functions providing the authorized administrator the capability to read all audit information from the audit records.

### FAU_SAR.2 Restricted audit review

This component satisfies **O.AUDIT** because the TSF ensures the functions prohibiting all users read access to the audit records except those the authorized administrators.

### FAU_SAR.3 Selectable audit review

This component satisfies **O.AUDIT** because the TSF ensures the functions providing the ability to perform searching and sorting of audit data.

### FAU_SEL.1 Selective audit

This component satisfies **O.AUDIT** because the TSF ensures the functions including or excluding auditable events from the set of audited events.

**FAU_STG.1 Protected audit trail storage**

This component satisfies **O.AUDIT** because the TSF ensures the functions protecting the stored audit records from the unauthorized deletion and modification.

**FAU_STG.3 Action in case of possible audit data loss**

This component satisfies **O.AUDIT** because the TSF ensures the functions taking the corresponding actions if the audit trail exceeds the audit space defined by the authorized administrator.

**FAU_STG.4 Prevention of audit data loss**

This component satisfies **O.AUDIT** because the TSF ensures the functions preventing auditable events and taking the corresponding actions if the audit trail is full.

**FDP_ACC.1 Subset access control**

This component satisfies **O.DAC** because the TSF ensures the functions enforcing the Discretionary Access Control Policy on all processes.

**FDP_ACF.1 Security attributes based access control**

This component satisfies **O.DAC** because the TSF ensures the functions enforcing the Discretionary Access Control Policy to objects based on attributes.

**FDP_IFC.1 Subset information flow control**

This component satisfies **O.MAC** because the TSF ensures the functions enforcing the Mandatory Access Control Policy on all subjects.

### FDP_IFF.2 Hierarchical security attributes

This component satisfies **O.MAC** because the TSF ensures the functions enforcing the Mandatory Access Control Policy based on the security label of subject and object.

### FDP_ITC.1 Import of user data without security attributes

This component satisfies **O.MAC** because the TSF ensures the functions enforcing the Mandatory Access Control Policy when importing user data from outside of the TSC.

### FDP_RIP.1 Subset residual information protection

This component satisfies **O.RESIDUE** because the TSF ensures the functions ensuring that any previous information content of a resource is made unavailable.

### FIA_AFL.1 Authentication failure handling

This component satisfies **O.IA** because the TSF ensures the functions detecting when 5 unsuccessful authentication attempts occur.

### FIA_ATD.1 User attributes definition

This component satisfies **O.IA** because the TSF ensures the functions maintaining the list of security attributes belonging to individual users.

### FIA_SOS.1 Verification of secrets

This component satisfies **O.IA** because the TSF ensures the functions providing a mechanism to verify that secrets meet the defined quality metric.

### FIA_UAU.1 Timing of authentication

This component satisfies **O.IA** because the TSF ensures the functions allowing the list of TSF mediated actions on behalf of the user to be performed before the user is authenticated.

**FIA_UAU.4 Single-use authentication mechanisms**

This component satisfies **O.IA** because the TSF ensures the functions preventing reuse of authentication data related to the authentication mechanism.

**FIA_UAU.7 Protected authentication feedback**

This component satisfies **O.IA** because the TSF ensures the functions providing only '*' or 'space' to the user while the authentication is in progress.

**FIA_UID.2 User identification before any action**

This component satisfies **O.MANAGE, O.INTEGRITY, and O.IA** because the TSF ensures the functions identifying each user before allowing any other TSF-mediated actions on behalf of that user.

**FIA_USB.1 User-subject binding**

This component satisfies **O.IA** because the TSF ensures the functions associating user security attributes with subjects acting on behalf of that user.

**FMT_MOF.1 Management of security functions behavior**

This component satisfies **O.MANAGE** because the TSF ensures to be managed the functions by the authorized administrator only.

**FMT_MSA.1(1) Management of security attributes**

This component satisfies **O.MANAGE and O.DAC** because the TSF ensures to be managed the DAC policy by the authorized administrator only.

### FMT_MSA.1(2) Management of security attributes

This component satisfies **O.MANAGE, O.LABEL and O.MAC** because the TSF ensures to be managed the security label for the MAC policy by the authorized administrator only.

### FMT_MSA.3(1) Static attributes initialization

This component satisfies **O.MANAGE** and **O.DAC** because the TSF ensures the functions providing the restrictive default values for security attributes to be used by enforcing the DAC policy.

### FMT_MSA.3(2) Static attributes initialization

This component satisfies **O.MANAGE, O.LABEL and O.MAC** because the TSF ensures the functions providing the restrictive default values for security attributes to be applied by enforcing the MAC policy.

### FMT_MTD.1(1) Management of TSF data

This component satisfies **O.MANAGE** because the TSF ensures the functions restricting the ability to change, query, delete, and clear the audit data to the authorized administrator only.

### FMT_MTD.1(2) Management of TSF data

This component satisfies **O.MANAGE** because the TSF ensures the functions restricting the ability to delete, and initialize the identification and authentication data to the authorized administrator only.

### FMT_MTD.1(3) Management of TSF data

This component satisfies **O.MANAGE** because the TSF ensures the functions restricting the ability to modify the authentication data to the authorized administrator or the owner of data.

### FMT_MTD.1(4) Management of TSF data

This component satisfies **O.MANAGE** because the TSF ensures the functions restricting the ability to change, query, delete, clear, and create the TSF data associated with security to the authorized administrator only.

### FMT_REV.1(1) Revocation

This component satisfies **O.MANAGE and O.LABEL** because the TSF ensures the functions restricting the ability to revoke security attributes associated with the users within the TSC to the authorized administrator only.

### FMT_REV.1(2) Revocation

This component satisfies **O.MANAGE and O.LABEL** because the TSF ensures the functions restricting the ability to revoke security attributes associated with the objects within the TSC to the authorized administrator only.

### FMT_SMF.1 Specification of Management Functions

This component satisfies **O.MANAGE** because the TSF ensures the management functions of security attributes, security function and so on.

### FMT_SMR.1 Security roles

This component satisfies **O.MANAGE** because the TSF ensures the functions maintaining the user roles and associating users with roles.

### FPT_AMT.1 Abstract machine testing

This component satisfies **O.INTEGRITY and O.PROTECT** because the TSF ensures the functions running a suit of tests to demonstrate the correct operation.

### FPT_ITT.1 Basic internal TSF data transfer protection

This component satisfies **O.MANAGE** because the TSF ensures the functions protecting TSF data from disclosure and modification when it is transmitted between separate parts of the TOE.

### FPT_RVM.1 TSP Non-bypass ability of the TSP

This component satisfies **O.PROTECT** because the TSF ensures the functions invoking and succeeding the TSP enforcement functions.

### FPT_SEP.1 TSF domain separation

This component satisfies **O.PROTECT** because the TSF ensures the functions maintaining a security domain for its own execution that protects it from interference and tampering by distrusted subjects.

### FPT_STM.1 Reliable time stamps

This component satisfies **O.AUDIT** because the TSF ensures the functions providing reliable time stamps for its own use.

### FPT_TST.1 TSF testing

This component satisfies **O.INTEGRITY and O.PROTECT** because the TSF ensures the functions running a suite of self tests to demonstrate the correct operation of the TSF and providing the authorized users with the capability to verify the integrity of the TSF data and executable code.

### FTA_SSL.1 TSF-initiated session locking

This component satisfies **O.INTEGRITY and O.PROTECT** because the TSF ensures the functions locking an interactive session after time interval of administrator inactivity.

### FTP_ITC.1 Inter-TSF trusted channel

This component satisfies **O.MANAGE** because the TSF ensures the functions providing a communication channel between itself and a remote trusted IT product and providing assured identification of its end points.

### 8.2.2 Rationale for TOE Assurance Requirements

The assurance requirements of this ST are consist of the one of Common Criteria Part 3, and the target evaluation assurance level for the product is EAL3+. The augmented components in this ST are listed as below.

- ➢ ADV_IMP.2 Implementation of the TSF
- ➢ ADV_LLD.1 Descriptive low-level design
- ➢ ALC_TAT.1 Well-defined development tools
- ➢ ATE_DPT.2 Testing: low-level design
- ➢ AVA_VLA.2 Independent vulnerability analysis

The component **ADV_IMP.2** and **ATE_DPT.2** are augmented by the Protection Profile because the security objective **O.CODE** is required.

For dependency with **ADV_IMP.2**, the component **ADV_LLD.1** and **ALC_TAT.1** are augmented by the Protection Profile.

The component **AVA_VLA.2** is augmented by the Protection Profile because the vulnerability analysis should be performed not only by developer but also by evaluator.

### 8.2.3 Rationale for the IT environment requirements

**[Table 8-4] Responding Requirements for ST and IT Environment**

| ST<br>Security<br>Function Requirements | OE.SSL protocol | OE.TIME |
|---|---|---|

| | | |
|---|---|---|
| FPT_STM.1 | | X |
| FPT_ITT.1 | X | |
| FTP_ITC.1 | X | |

**FPT_STM.1 Reliable time stamps**

This component satisfies **OE.TIME** because it is assumed that IT Environment (Operating System) provides the reliable timestamp, which is used by the TSF.

**FPT_ITT.1 Basic internal TSF data transfer protection**

This component satisfies **OE.SSL** because it is assumed that IT Environment (SSL protocol) provides the secure communication between separate parts of the TOE.

**FTP_ITC.1 Inter-TSF trusted channel**

This component satisfies **OE.SSL** because it is assumed that IT Environment (SSL protocol) provides the secure communication between separate parts of the TOE.

## 8.3 Rationale for Dependencies

### 8.3.1 Dependencies between security functions

The following table shows dependency of functional components.

FAU_GEN.2, FIA_UAU.1, FMT_SMR.1 have dependency on the FIA_UID.1 and this is satisfied by FIA_UID.2 which has hierarchical relationship with FIA_UID.1.

FDP_IFC.1 has dependency on the FDP_IFF.1 and this is satisfied by FDP_IFF.2 which has hierarchical relationship with FDP_IFF.1.

**[Table 8-5] Functional Component Dependency**

| No | Functional component | Dependency | Reference No. |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1 | FAU_ARP.1 | FAU_SAA.1 | 4 |
| 2 | FAU_GEN.1 | FPT_STM.1 | 43 |
| 3 | FAU_GEN.2 | FAU_GEN.1, FIA_UID.1 | 2, 25 |
| 4 | FAU_SAA.1 | FAU_GEN.1 | 2 |
| 5 | FAU_SAA.3 | - | - |
| 6 | FAU_SAR.1 | FAU_GEN.1 | 2 |
| 7 | FAU_SAR.2 | FAU_SAR.1 | 6 |
| 8 | FAU_SAR.3 | FAU_SAR.1 | 6 |
| 9 | FAU_SEL.1 | FAU_GEN.1<br>FMT_MTD.1 | 2<br>32, 33, 34, 35 |
| 10 | FAU_STG.1 | FAU_GEN.1 | 2 |
| 11 | FAU_STG.3 | FAU_STG.1 | 10 |
| 12 | FAU_STG.4 | FAU_STG.1 | 10 |
| 13 | FDP_ACC.1 | FDP_ACF.1 | 14 |
| 14 | FDP_ACF.1 | FDP_ACC.1, FMT_MSA.3 | 13, 30 |
| 15 | FDP_IFC.1 | FDP_IFF.1 | 16 |
| 16 | FDP_IFF.2 | FDP_IFC.1, FMT_MSA.3 | 15, 31 |
| 17 | FDP_ITC.1 | [FDP_IFC.1], FMT_MSA.3 | 15, 31 |
| 18 | FDP_RIP.1 | - | - |
| 19 | FIA_AFL.1 | FIA_UAU.1 | 22 |
| 20 | FIA_ATD.1 | - | - |
| 21 | FIA_SOS.1 | - | - |
| 22 | FIA_UAU.1 | FIA_UID.1 | 25 |
| 23 | FIA_UAU.4 | - | - |
| 24 | FIA_UAU.7 | FIA_UAU.1 | 22 |
| 25 | FIA_UID.2 | - | - |
| 26 | FIA_USB.1 | FIA_ATD.1 | 20 |
| 27 | FMT_MOF.1 | FMT_SMF.1, FMT_SMR.1 | 38, 39 |
| 28 | FMT_MSA.1(1) | [FDP_ACC.1],<br>FMT_SMF.1, FMT_SMR.1 | 13<br>38, 39 |
| 29 | FMT_MSA.1(2) | [FDP_IFC.1],<br>FMT_SMF.1, FMT_SMR.1 | 15<br>38, 39 |
| 30 | FMT_MSA.3(1) | FMT_MSA.1, FMT_SMR.1 | 28, 39 |
| 31 | FMT_MSA.3(2) | FMT_MSA.1, FMT_SMR.1 | 29, 39 |

| | | | |
|---|---|---|---|
| 32 | FMT_MTD.1(1) | FMT_SMF.1, FMT_SMR.1 | 38, 39 |
| 33 | FMT_MTD.1(2) | FMT_SMF.1, FMT_SMR.1 | 38, 39 |
| 34 | FMT_MTD.1(3) | FMT_SMF.1, FMT_SMR.1 | 38, 39 |
| 35 | FMT_MTD.1(4) | FMT_SMF.1, FMT_SMR.1 | 38, 39 |
| 36 | FMT_REV.1(1) | FMT_SMR.1 | 39 |
| 37 | FMT_REV.1(2) | FMT_SMR.1 | 39 |
| 38 | FMT_SMF.1 | - | - |
| 39 | FMT_SMR.1 | FIA_UID.1 | 25 |
| 40 | FPT_AMT.1 | - | - |
| 47 | FPT_ITT.1 | - | - |
| 41 | FPT_RVM.1 | - | - |
| 42 | FPT_SEP.1 | - | - |
| 43 | FPT_STM.1 | - | - |
| 44 | FPT_TST.1 | FPT_AMT.1 | 40 |
| 45 | FTA_SSL.1 | FIA_UAU.1 | 22 |
| 46 | FTP_ITC.1 | - | - |

### 8.3.2 Dependencies between assurance requirements

The following table shows that the dependencies between the components of security functional requirements.

ALC_TAT.1, AVA_VLA.2 has dependency on ADV_IMP.1 and this is satisfied by ADV_IMP.2 which has hierarchical relationship with ADV_IMP.1.

**[Table 8-6] Dependencies between assurance components added**

| No | Functional component | Dependency | Reference No. |
|---|---|---|---|
| 1 | ADV_IMP.2 | ADV_LLD.1<br>ADV_RCR.1<br>ALC_TAT.1 | 2<br>EAL3<br>3 |
| 2 | ADV_LLD.1 | ADV_HLD.2<br>ADV_RCR.1 | EAL3<br>EAL3 |
| 3 | ALC_TAT.1 | ADV_IMP.1 | 1 |

| | | | |
|---|---|---|---|
| 4 | ATE_DPT.2 | ADV_HLD.2<br>ADV_LLD.1<br>ATE_FUN.1 | EAL3<br>2<br>EAL3 |
| 5 | AVA_VLA.2 | ADV_FSP.1<br>ADV_HLD.2<br>ADV_IMP.1<br>ADV_LLD.1<br>AGD_ADM.1<br>AGD_USR.1 | EAL3<br>EAL3<br>1<br>2<br>EAL3<br>EAL3 |

## 8.4 TOE Summary Specification Rationale

### 8.4.1 Conformance to TOE Security Functions

The following are security function list described in the TOE Summary Specification.

**[Table 8-7] TOE Security Functions**

| ID | Security Function | ID | Security Function |
|---|---|---|---|
| Refer.1 | System Call Interception | Admin.2 | Hierarchical Category Management |
| Refer.2 | Security Module Separation | Admin.3 | Labeled Users Management |
| Ac_mac.1 | Security Label Assignment | Admin.4 | Labeled Objects Management |
| Ac_mac.2 | Multi-Level based MAC | Admin.5 | Labeled Processes Management |
| Ac_mac.3 | Inheritance and Revocation | Admin.6 | ACL Policies Management |
| Ac_dac.1 | ACL based DAC | Admin.7 | Allowed/Denied List Management |
| Ac_dac.2 | Allowed/Denied List based DAC | Admin.8 | Audit Configuration |
| Auth.1 | ESM Authentication | Admin.9 | ESM Users Management |
| Auth.2 | SecureOS Authentication | Admin.10 | Security Password Management |
| Audit.1 | Audit Generation and Collection | Admin.11 | Security Functions Configuration |
| Audit.2 | Potential Violation Analysis | Admin.12 | System Services Management |
| Audit.3 | Audit Storage Management | Protect.1 | Abstract Machine Testing |
| Audit.4 | Audit Review | Protect.2 | Integrity Functions |
| Audit.5 | Simple Attack Prevention | Protect.3 | ESM Screen Saving |
| Admin.1 | Security Functions Management | Protect.4 | Secure Communication |

The following table shows that the IT security functions, as specified in the TOE summary specification, meet all security functional requirements for the TOE and work together to satisfy the TOE security functional requirements.

**[Table 8-8] Mapping TOE SFR to TOE SS**

| TOE Security Functional Requirements | | | TOE Summary Specification |
|---|---|---|---|
| Class | Component | Element | |
| Security Audit | FAU_ARP.1 | FAU_ARP.1.1 | Audit.2 |
| | FAU_GEN.1 | FAU_GEN.1.1 | Audit.1 |
| | | FAU_GEN.1.2 | Audit.1 |
| | FAU_GEN.2 | FAU_GEN.2.1 | Audit.1 |
| | FAU_SAA.1 | FAU_SAA.1.1 | Audit.2 |
| | | FAU_SAA.1.2 | Audit.2 |
| | FAU_SAA.3 | FAU_SAA.3.1 | Audit.5 |
| | | FAU_SAA.3.2 | Audit.5 |
| | | FAU_SAA.3.3 | Audit.5 |
| | FAU_SAR.1 | FAU_SAR.1.1 | Audit.4 |
| | | FAU_SAR.1.2 | Audit.4 |
| | FAU_SAR.2 | FAU_SAR.2.1 | Audit.4 |
| | FAU_SAR.3 | FAU_SAR.3.1 | Audit.4 |
| | FAU_SEL.1 | FAU_SEL.1.1 | Audit.1 |
| | FAU_STG.1 | FAU_STG.1.1 | Audit.3 |
| | | FAU_STG.1.2 | Audit.3 |
| | FAU_STG.3 | FAU_STG.3.1 | Audit.3 |
| | FAU_STG.4 | FAU_STG.4.1 | Audit.3 |
| User Data Protection | FDP_ACC.1 | FDP_ACC.1.1 | Ac_dac.1, Ac_dac.2 |
| | FDP_ACF.1 | FDP_ACF.1.1 | Ac_dac.1 |
| | | FDP_ACF.1.2 | Ac_dac.1 |
| | | FDP_ACF.1.3 | Ac_dac.2 |
| | | FDP_ACF.1.4 | Ac_dac.2 |
| | FDP_IFC.1 | FDP_IFC.1.1 | Ac_mac.1, Ac_mac.2 |
| | FDP_IFF.2 | FDP_IFF.2.1 | Ac_mac.1, Ac_mac.2 |
| | | FDP_IFF.2.2 | Ac_mac.1, Ac_mac.2 |
| | | FDP_IFF.2.3 | Ac_mac.1, Ac_mac.2 |

| | | | |
|---|---|---|---|
| | | FDP_IFF.2.4 | Ac_mac.1, Ac_mac.2 |
| | | FDP_IFF.2.5 | Ac_mac.1, Ac_mac.2 |
| | | FDP_IFF.2.6 | Ac_mac.1, Ac_mac.2 |
| | | FDP_IFF.2.7 | Ac_mac.2 |
| | FDP_ITC.1 | FDP_ITC.1.1 | Ac_mac.2, Ac_mac.3 |
| | | FDP_ITC.1.2 | Ac_mac.2, Ac_mac.3 |
| | | FDP_ITC.1.3 | Ac_mac.3 |
| | FDP_RIP.1 | FDP_RIP.1.1 | Ac_mac.3 |
| IA | FIA_AFL.1 | FIA_AFL.1.1 | Auth.1, Auth.2 |
| | | FIA_AFL.1.2 | Auth.1, Auth.2 |
| | FIA_ATD.1 | FIA_ATD.1.1 | Admin.3 |
| | FIA_SOS.1 | FIA_SOS.1.1 | Admin.9, Admin.10 |
| | FIA_UAU.1 | FIA_UAU.1.1 | Auth.1, Auth.2 |
| | | FIA_UAU.1.2 | Auth.1, Auth.2 |
| | FIA_UAU.4 | FIA_UAU.4.1 | Auth.2, Protect.4 |
| | FIA_UAU.7 | FIA_UAU.7.1 | Auth.1, Auth.2 |
| | FIA_UID.2 | FIA_UID.2.1 | Auth.2 |
| | FIA_USB.1 | FIA_USB.1.1 | Ac_mac.1 |
| | | FIA_USB.1.2 | Ac_mac.1 |
| | | FIA_USB.1.3 | Ac_mac.1 |
| Security Management | FMT_MOF.1 | FMT_MOF.1.1 | Admin.1 |
| | FMT_MSA.1(1) | FMT_MSA.1.1 | Admin.6, Admin.7 |
| | FMT_MSA.1(2) | FMT_MSA.1.1 | Admin.2, Admin.3, Admin.4, Admin.5 |
| | FMT_MSA.3(1) | FMT_MSA.3.1 | Admin.6, Admin.7 |
| | | FMT_MSA.3.2 | Admin.6, Admin.7 |
| | FMT_MSA.3(2) | FMT_MSA.3.1 | Admin.2, Admin.3, Admin.4 |
| | | FMT_MSA.3.2 | Admin.2, Admin.3, Admin.4 |
| | FMT_MTD.1(1) | FMT_MTD.1.1 | Admin.8 |
| | FMT_MTD.1(2) | FMT_MTD.1.1 | Admin.3, Admin.9 |
| | FMT_MTD.1(3) | FMT_MTD.1.1 | Admin.9, Admin.10 |
| | FMT_MTD.1(4) | FMT_MTD.1.1 | Admin.11, Protect.2, Protect.3 |
| | FMT_REV.1(1) | FMT_REV.1.1 | Admin.3, Admin.9 |
| | | FMT_REV.1.2 | Admin.3, Admin.9 |

| | | | |
|---|---|---|---|
| | FMT_REV.1(2) | FMT_REV.1.1<br>FMT_REV.1.2 | Admin.4<br>Admin.4 |
| | FMT_SMF.1 | FMT_SMF.1.1 | Admin.1, Admin.2, Admin.3, Admin.4, Admin.5, Admin.6, Admin.7, Admin.8, Admin.9, Admin.10, Admin.11, Admin.12, Protect.1, Protect.2, Protect.3 |
| | FMT_SMR.1 | FMT_SMR.1.1 | Admin.1, Admin.2, Admin.3, Admin.4, Admin.5, Admin.6, Admin.7, Admin.8, Admin.9, Admin.10, Admin.11, Admin.12 |
| | | FMT_SMR.1.2 | Admin.1, Admin.2, Admin.3, Admin.4, Admin.5, Admin.6, Admin.7, Admin.8, Admin.9, Admin.10, Admin.11, Admin.12 |
| TSF Protection | FPT_AMT.1 | FPT_AMT.1.1 | Protect.1 |
| | FPT_ITT.1 | FPT_ITT.1.1 | Protect.4 |
| | FPT_RVM.1 | FPT_RVM.1.1 | Refer.1 |
| | FPT_SEP.1 | FPT_SEP.1.1<br>FPT_SEP.1.2 | Refer.2<br>Refer.2 |
| | FPT_STM.1 | FPT_STM.1.1 | Audit.1 |
| | FPT_TST.1 | FPT_TST.1.1<br>FPT_TST.1.2<br>FPT_TST.1.3 | Protect.1, Protect.2<br>Protect.2<br>Protect.2 |
| TOE Access | FTA_SSL.1 | FTA_SSL.1.1<br>FTA_SSL.1.2 | Protect.3<br>Protect.3 |
| Trusted Path/Channels | FTP_ITC.1 | FTP_ITC.1.1<br>FTP_ITC.1.2<br>FTP_ITC.1.3 | Protect.4<br>Protect.4<br>Protect.4 |

### Refer.1 System Call Interception

This satisfies the **FPT_RVM.1 Non-bypass ability** because the Refer.1 ensures that all controlled system calls are intercepted.

### Refer.2 Security Module Separation

This satisfies the **FPT_SEP.1 TSF Domain Separation** because the Refer.2 maintains a secure domain within the commercial operating system for trusted execution.

### Ac_mac.1 Security Label Assignment

This satisfies the **FDP_IFC.1 Subset Information Flow Control** because the Ac_mac.1 applies MAC policy between creating subject and created subject when performing subject creation on behalf of the user.

This satisfies the **FDP_IFF.2 Hierarchical Security Attributes** because the Ac_mac.1 provides the functions applying policies of multi-level based MAC when new subject is created, and configuring security attributes for subject in case of allowed enforcement.

This satisfies the **FIA_USB.1 User-Subject Binding** because the Ac_mac.1 provides the functions configuring security attributes on the basis of security attributes of users for performing subject on behalf of the user.

### Ac_mac.2 Multi-Level based MAC

This satisfies the **FDP_IFC.1 Subset Information Flow Control** because the Ac_mac.2 applies MAC policy over mandatory operation when a subject accesses to an object.

This satisfies the **FDP_IFF.2 Hierarchical Security Attributes** because the Ac_mac.2 applies MAC by multi-level based security attributes of subject and object when a subject accesses to an object.

This satisfies the **FDP_ITC.1 Import of User Data without Security Attributes** because the Ac_mac.2 ensures the functions enforcing the MAC policy for importing user data without security attributes.

### Ac_mac.3 Inheritance and Revocation

This satisfies the **FDP_ITC.1 Import of User Data without Security Attributes** because the Ac_mac.3 inherits security attributes based on subject for the imported data from outside of the TSC.

This satisfies the **FDP_RIP.1 Subset Residual Information Protection** because the Ac_mac.3 provides the functions to inherit attributes of subject when an object is created, and revokes security attributes and previous information content when an object is deleted to ensure that the previous information of a resource is no longer available.

**Ac_dac.1 ACL based DAC**

This satisfies the **FDP_ACC.1 Subset Access Control** because the Ac_dac.1 provides the functions enforcing DAC based on ACL policies when a subject accesses to an object through operation of DAC.

This satisfies the **FDP_ACF.1 Security Attributes based Access Control** because the Ac_dac.1 provides the functions enforcing DAC based on ACL policy including the security attributes of subject(identity of subject, group memberships of subject, process of subject) and the security attributes of object(access permission operation).

**Ac_dac.2 Allow/Deny List based DAC**

This satisfies the **FDP_ACC.1 Subset Access Control** because the Ac_dac.2 provides the functions enforcing DAC based on the allowed rule of setuid, the allowed rule for su operation, the denied rule of the restricting command execution, the denied rule of the controlling kill signal when a subject try to access to an object through the DAC operation.

This satisfies the **FDP_ACF.1 Security Attributes based Access Control** because the Ac_dac.2 provides the function enforcing DAC on the basis of security attributes of subject (security role status of subject).

**Auth.1 ESM Authentication**

This satisfies the **FIA_AFL.1 Authentication Failure Handling** because the Auth.1 provides the function to terminate the ESM if ESM's authentication of authorized administrator fails 5 times in succession.

This satisfies the **FIA_UAU.1 Timing of Authentication** because the Auth.1 provides user authentication function for ESM.

This satisfies the **FIA_UAU.7 Protected Authentication Feedback** because the Auth.1 provides only '*' or space by appointed authentication feedback to the user of ESM while the authentication is in progress.

**Auth.2 SecureOS Authentication**

This satisfies the **FIA_AFL.1 Authentication Failure Handling** because the Auth.2 provides the function to disconnect with Secure OS if the authentication of authorized administrator fails 5 times in succession.

This satisfies the **FIA_UAU.1 Timing of Authentication** because the Auth.2 provides user authentication function for SecureOS.

This satisfies the **FIA_UAU.4 Single-Use Authentication Mechanisms** because the Auth.2 ensures that SecureOS authentication data can not be reused by using SSL protocol.

This satisfies the **FIA_UAU.7 Protected Authentication Feedback** because the Auth.2 provides only '*' or space by appointed authentication feedback to the user while the authentication is in progress.

This satisfies the **FIA_UID.2 User Identification before any Action** because the Auth.2 ensures the function identifying whether a connection is allowed on the basis of administrator's identification and ESM's connection IP before authenticating an administrator.

**Audit.1 Audit Data Generation and Collection**

This satisfies the **FAU_GEN.1 Audit Data Generation** because the Audit.1 provides function generating an audit record for the auditable events.

This satisfies the **FAU_GEN.2 User Identity Association** because the Audit.1 ensures the function associating each auditable event with the identity of the user that caused the event.

This satisfies the **FAU_SEL.1 Selective Audit** because the Audit.1 ensures the function including or excluding auditable events from the set of audited events.

This satisfies the **FPT_STM.1 Reliable Time Stamps** because the Audit.1 provides reliable time stamps requiring at audit generation and collection in sequence from the ESM.

### Audit.2 Potential Violation Analysis

This satisfies the **FAU_ARP.1 Security Alarms** because the Audit.2 provides corresponding action in case of potential violation detection.

This satisfies the **FAU_SAA.1 Potential Violation Analysis** because the Audit.2 ensures the function indicating as a potential violation when accumulation and combination of security violation attained to the configured value.

### Audit.3 Audit Data Storage Management

This satisfies the **FAU_STG.1 Protected Audit Trail Storage** because the Audit.3 provides the functions protecting the stored audit records from the unauthorized deletion and modification.

This satisfies the **FAU_STG.3 Action in case of Possible Audit Data Loss** because the Audit.3 ensures the protection functions of possible audit data loss taking the corresponding actions notifying to administrator if the audit trail exceeds the audit space defined by the authorized administrator.

This satisfies the **FAU_STG.4 Prevention of Audit Data Loss** because the Audit.3 provides the preventive function of the auditable events deferring the TSF to be called by all users except the authorized administrator if the audit trail is full.

### Audit.4 Audit Data Review

This satisfies the **FAU_SAR.1 Audit Review** because the Audit.4 provides function that authorized administrator is able to review audit data.

This satisfies the **FAU_SAR.2 Restricted Audit Review** because the Audit.4 ensures the function prohibiting all users read access to the audit records except those the authorized administrators.

This satisfies the **FAU_SAR.3 Selectable Audit Review** because the Audit.4 ensures the function providing the ability to perform searching and sorting of audit data.

### Audit.5 Simple Attack Prevention

This satisfies the **FAU_SAA.3 Simple Attack Heuristics** because the Audit.5 provides the functions detecting the signature events from the auditable events and indicating a potential violation.

### Admin.1 Security Functions Management

This satisfies the **FMT_MOF.1 Management of Security Functions Behavior** because the Admin.1 provides functions that authorized administrator is able to start or stop security functions.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.1 provides GUI and CLI to start or stop security functions.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.1 ensures that associating a user with the role of authorized administrator.

### Admin.2 Hierarchical Category Management

This satisfies the **FMT_MSA.1(2) Management of Security Attributes (MAC)** because the Admin.2 provides the function that authorized administrator manages non-hierarchical category.

This satisfies the **FMT_MSA.3(2) Static Attributes Initialization (MAC)** because the Admin.2 provides initial value of non-hierarchical category.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.2 provides GUI and CLI to manage security category.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.2 ensures that associating a user with the role of authorized administrator.

### Admin.3 Labeled Users Management

This satisfies the **FIA_ATD.1 User Attributes Definition** because the Admin.3 defines security attributes of labeled users.

This satisfies the **FMT_MSA.1(2) Management of Security Attributes (MAC)** because the Admin.3 provides functions that authorized administrator can assign, modify, and delete the security attributes of users.

This satisfies the **FMT_MSA.3(2) Static Attributes Initialization (MAC)** because the Admin.3 provides initial value when an authorized administrator assigns security attributes to users.

This satisfies the **FMT_MTD.1(2) Management of TSF Data (Identification & Authentication)** because the Admin.3 provides the functions initializing and deleting audit data of this user when an authorized administrator assigns or revokes security attributes of user.

This satisfies the **FMT_REV.1(1) Revocation (User)** because the Admin.3 assigns ability to revoke security attributes of labeled users to the authorized user only.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.3 provides GUI and CLI to manage security attributes of labeled user.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.3 ensures associating a user with the role of authorized administrator.

### Admin.4 Labeled Objects Management

This satisfies the **FMT_MSA.1(2) Management of Security Attributes (MAC)** because the Admin.4 provides functions that authorized administrator can assign and delete security attributes of object(files).

This satisfies the **FMT_MSA.3(2) Static Attributes Initialization (MAC)** because the Admin.4 provides initial value when an authorized administrator assigns security attributes to object(files).

This satisfies the **FMT_REV.1(2) Revocation (Object)** because the Admin.4 ensures an exclusive ability for the authorized administrator to revoke security attributes of the object(files).

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.4 provides GUI and CLI to manage security attributes of object (file).

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.4 ensures associating a user with the role of authorized administrator.

### Admin.5 Labeled Processes Management

This satisfies the **FMT_MSA.1(2) Management of Security Attributes (MAC)** because the Admin.5 provides the function that authorized administrator can query the security attributes of subject(processes).

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.5 provides GUI and CLI to query security attributes of subject.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.5 ensures associating a user with the role of authorized administrator.

### Admin.6 ACL Policies Management

This satisfies the **FMT_MSA.1(1) Management of Security Attributes (DAC)** because the Admin.6 provides functions that authorized administrator query, add, modify, and delete policies of ACL based DAC.

This satisfies the **FMT_MSA.3(1) Static Attributes Initialization (DAC)** because the Admin.6 provides initial value when an authorized administrator adds ACL policy.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.6 provides GUI and CLI to manage ACL policies.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.6 ensures associating a user with the role of authorized administrator.

**Admin.7 Allow/Deny List Management**

This satisfies the **FMT_MSA.1(1) Management of Security Attributes (DAC)** because the Admin.7 provides functions that authorized administrator can query, add, and delete allowed/denied list of DAC policies.

This satisfies the **FMT_MSA.3(1) Static Attributes Initialization (DAC)** because the Admin.7 provides initial value when an authorized administrator adds allowed/denied list.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.7 provides GUI and CLI to manage allowed/denied list.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.7 ensures associating a user with the role of authorized administrator.

**Admin.8 Audit Data Configuration**

This satisfies the **FMT_MTD.1(1) Management of TSF Data (Audit Data)** because the Admin.8 provides ability that authorized administrator can manage the configuration of audit data environment and alarms.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.8 provides GUI and CLI to manage the configuration of audit data environment and alarms.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.8 ensures associating a user with the role of authorized administrator.

**Admin.9 ESM User Management**

This satisfies the **FIA_SOS.1 Verification of Secrets** because the Admin.9 ensures the functions providing a mechanism to verify whether the ESM user's registered password satisfies the defined quality criteria.

This satisfies the **FMT_MTD.1(2) Management of TSF Data (Identification and Authentication Data)** because the Admin.9 provides the function to delete authentication data of the user when the ESM user is deleted.

This satisfies the **FMT_MTD.1(3) Management of TSF Data (Authentication Data)** because the Admin.9 provides the function that only authorized administrator and allowed user can modify identification and authentication data of ESM users.

This satisfies the **FMT_REV.1(1) Revocation (User)** because the Admin.9 assigns revocable ability of administrator to authorized user.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.9 provides GUI to manage user of ESM.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.9 ensures associating a user with the role of authorized administrator.

**Admin.10 Security Password Management**

This satisfies the **FIA_SOS.1 Verification of Secrets** because the Admin.10 ensures the functions providing a mechanism to verify whether the Secure OS user's registering security password satisfies the defined quality.

This satisfies the **FMT_MTD.1(3) Management of TSF Data (Authentication Data)** because the Admin.10 provides ability that only authorized administrator and allowed user can modify identification and authentication data of SecureOS users.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.10 provides GUI to manage Secure OS users.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.10 ensures associating a user with the role of authorized administrator.

**Admin.11 Security Functions Configuration**

This satisfies the **FMT_MTD.1(4) Management of TSF Data (TSF Data)** because the Admin.11 provides the function for an authorized administrator to manage operating environment of security functions.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.11 provides GUI and CLI to configure operating environment of security functions.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.11 ensures associating a user with the role of authorized administrator.

### Admin.12 System Services Management

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Admin.12 provides GUI of system services that is able to configure and to manage policies of IP Filter, system monitoring, and system account management existing outside of the TOE.

This satisfies the **FMT_SMR.1 Security Roles** because the Admin.12 ensures associating a user with the role of authorized administrator.

### Protect.1 Abstract Machine Testing

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Protect.1 provides GUI to indicate the result of abstract machine testing.

This satisfies the **FPT_AMT.1 Abstract Machine Testing** because the Protect.1 ensures the functions running a suite of tests when administrator requests during start-up, periodically during operation to demonstrate the correct operation of the abstract machine that underlies the TSF.

This satisfies the **FPT_TST.1 TSF Testing** because the Protect.1 ensures the functions running a suite of self tests when administrator requests during start-up, periodically during operation to demonstrate the correct operation of the TSF.

### Protect.2 Integrity Functions

This satisfies the **FMT_MTD.1(4) Management of TSF Data (TSF Data)** because the Protect.2 provides the function for an authorized administrator to manage target items and performance of integrity check.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Protect.2 provides GUI to perform integrity check and managing target items.

This satisfies the **FPT_TST.1 TSF Testing** because the Protect.2 ensures the functions performing integrity checking for executable files during start-up, periodically during operation to demonstrate the correct operation of the TSF and providing the authorized users with the capability to verify the integrity of the TSF data and executable code.

**Protect.3 ESM Screen Saving**

This satisfies the **FMT_MTD.1(4) Management of TSF Data (TSF Data)** because the Protect.3 provides the function for an authorized administrator to manage inactive time interval of ESM.

This satisfies the **FMT_SMF.1 Specification of Management Functions** because the Protect.3 provides GUI to manage inactive time interval of ESM.

This satisfies the **FTA_SSL.1 TSF-initiated Session Locking** because the Protect.3 ensures the functions locking an interactive session after inactive time interval of ESM user, and requiring events to initiate unlocking the session.

**Protect.4 Secure Communication**

This satisfies the **FIA_UAU.4 Single-use Authentication Mechanism** because the Protect.4 ensures preventing reuse of transmitted data by utilizing SSL version 3 protocol.

This satisfies the **FTP_ITC.1 Inter-TSF Trusted Channel** and the **FPT_ITT.1 Basic Internal TSF Data Transfer protection** because the Protect.4 ensures that secure channel is configured by SSL version 3 protocol, when an authorized administrator tries to manage TOE from a remote place.

### 8.4.2 Justification for Compliance of Assurance Measures

The TOE summary specification assurance rationale represents corresponding relationship between described assurance measure and the TOE security assurance requirement.

**[Table 8-9] Responding TOE Assurance Requirements**

| TOE Assurance Requirements | | | Summarized Specification Assurance |
|---|---|---|---|
| Class | Assurance Component | | |
| Configuration Management (ACM) | ACM_CAP.3 | Authorization Controls | Configuration Management Document |
| | ACM_SCP.1 | TOE CM Coverage | |
| Delivery & Operation (ADO) | ADO_DEL.1 | Delivery Procedure | Delivery Document |
| | ADO_IGS.1 | Installation, generation, and start-up procedures | Installation Guidance |
| Development (ADV) | ADV_FSP.1 | Informal Function Specifications | Functional Specification Document |
| | ADV_HLD.2 | Security enforcing high-level design | High Level Design |
| | ADV_IMP.2 | Implementation of the TSF | Implementation representation |
| | ADV_LLD.1 | Descriptive low-level design | Low Level Design |
| | ADV_RCR.1 | Informal correspondence demonstration | Correspondence information of the functional specification |
| Guidance (AGD) | AGD_ADM.1 | Administrator guidance | Admin Guidance |
| | AGD_USR.1 | User Guidance | User Guidance |
| Life cycle support | ALC_DVS.1 | Identification of security measures | Life cycle support document |
| | ALC_TAT.1 | Well-defined development tools | |
| Testing (ATE) | ATE_COV.2 | Analysis of coverage | Testing document |
| | ATE_DPT.2 | Testing: low-level design | |
| | ATE_FUN.1 | Functional testing | |

| | | | |
|---|---|---|---|
| | ATE_IND.2 | Independent testing-sample | N/A(evaluator) |
| Vulnerability analysis (AVA) | AVA_MSU.1 | Examination of guidance | Misuse analysis will be Included in the admin guidance and user guidance |
| | AVA_SOF.1 | Strength of TOE security function evaluation | Vulnerability analysis document |
| | AVA_VLA.2 | Independent vulnerability analysis | |

### Configuration Management

This assurance item satisfies **ACM_CAP.3 Authorization Controls** by providing control to assure not to occur any unauthorized modifications in the TOE and assuring appropriate functionality and usage of configuration management system.

This assurance item satisfies **ACM_SCP.1 TOE CM Coverage** for controlling modifications on configuration items under of control by the Configuration Management System only and assuring it can be modified by controlled methods according to the proper authorization.

### Delivery Document

This assurance item satisfies **ADO_DEL.1 Delivery Procedure** by describing developer requirements for detecting and preventing TOE modifications for delivery.

### Installation Guidance

This assurance item satisfies **ADO_IGS.1 Installation, Generation, and Start-up Procedures** by describing required procedures for safe installation, generation, and start-up of the TOE.

### Functional Specifications

This assurance item satisfies **ADV_FSP.1 Informal Specifications** by describing the TSF and the external interfaces of TSF in informal way consistently.

### High Level Design

This assurance item satisfies **ADV_HLD.2 Security Enforcing High-level Design** by describing the TSF architecture with its every subsystem and by separating TSP-conducting subsystem and other subsystems.

### Implementation Representations

This assurance item satisfies **ADV_IMP.2 Implementation of the TSF** by describing whole implementation of the TSF and co-relationship of all implemented parts with consistency.

### Low Level Design

This assurance item satisfies **ADV_LLD.1 Descriptive Low-level Design** by describing Low-level design of the TSF in informal way with consistency and identifying each TSP-performing module's objective, internal interfaces, and external interfaces by separating the TOE into TSP-performing and other modules.

### Rationale

This assurance item satisfies **ADV_RCR.1 Informal Correspondence Demonstration** by providing the conformance analysis between the TOE summarized specifications, functional specification, and implementation verified specification.

### Admin Guidance

This assurance item satisfies **AGD_ADM.1 Admin Guidance** by describing management function and its interfaces can be used by TOE administrator.

This assurance item satisfies **AVA_MSU.1 Examination of Guidance** by identifying all possible operation modes of the TOE, its influences and associated items for safe maintenance.

### User Guidance

This assurance item satisfies **AGD_USR.1 User Guidance** by describing management function and its interfaces can be used by TOE administrator.

This assurance item satisfies **AVA_MSU.1 Examination of Guidance** by identifying all possible operation modes of the TOE, its influences and associated items for safe maintenance.

### Life Cycle Support

This assurance item satisfies **ALC_DVS.1 Identification of Security Measures** by describing all physical, procedural, human and other security measures required to protect confidentiality and integrity of the TOE design and implementation process in development environment.

This assurance item satisfies **ALC_TAT.1 Well-defined Development Tools** by describing selection details of implementation depended select items of development tools.

### Testing Document

This assurance item satisfies **ATE_COV.2 Analysis of Coverage** by assuring complete conformance of TSF described in the functional specification with the test items identified in the test documents.

This assurance item satisfies **ATE_DPT.2 Testing : Low-level Design** by verifying the test items identified in the test documents are sufficient to prove the operation of TSF according to the High-level and Low-level design.

This assurance item satisfies **ATE_FUN.1 Functional Testing** by including the test plan, test procedure description, expected and actual test result.

**Vulnerability Analysis**

The assurance item satisfies **AVA_SOF.1 Strength of TOE Security Function Evaluation** by assuring the TOE security function strength is analyzed for each mechanism clarified and fulfills the functional strength allowed level defined in the Protection Profile/Security Target.

The assurance item satisfies **AVA_VLA.2 Independent Vulnerability Analysis** by assuring all identified vulnerabilities can't be exploited in the intended environment of TOE.

## 8.5 Rationale for Strength of Function

The information to be protected by this TOE is the general data of government and the value is medium. The assumed level of expertise of the attacker for all the threats is unsophisticated, with access to standard equipment and public information. In the Common Criteria[1)] recommend to provide the security functions higher than minimum SOF-basic against the threats which have the low level of the attack potential.

Therefore, this is done in accordance with the SOF-medium for the strength of function.

This TOE satisfies the SOF-medium for the following security functional requirements to meet the threats with **T.UAUTH and T.BYPASS**.

- FIA_UAU.1 Timing of authentication
- FIA_UAU.4 Single-use authentication mechanisms

This TOE satisfies the SOF-medium for the following security functional requirements to meet the security objectives with **O.IA**.

- FIA_UAU.1 Timing of authentication

[1)] In the Annex B.8 Strength of function and vulnerability analysis of the Common Evaluation Methodology for Information Technology Security, Part 2, the minimum strength of function based on the method of calculating the attack potential is defined.

- FIA_UAU.4 Single-use authentication mechanisms

This TOE satisfies the SOF-medium for the following security functional requirements.

- FIA_UAU.1 Timing of authentication
- FIA_UAU.4 Single-use authentication mechanisms