HiCommand Suite Common Component Security Target

May 10, 2007
Version 1.08
Hitachi, Ltd.

This document is a translation of the evaluated and certified security target written in Japanese.

HiCommand Suite Common Component Security Target

- Revision History -

No.	Date created/changed	ST version	Reason for change	Prepared by	Approved by
1	October 16, 2006	1.00	Initial version	Fujii, Hiraiwa	Ito
2	December 15, 2006	1.01	The comments contained in EVE-EOR-0001-00 and EVE-EOR-1101-00 were incorporated.	Fujii	Ito
3	February 9, 2007	1.02	Evaluator comments were incorporated.	Fujii, Hiraiwa	Ito
4	February 19, 2007	1.03	The comments contained in EVE-EOR-0002-00 to EVE-EOR-0007-00 incorporated.	Fujii	Ito
5	February 23, 2007	1.04	Evaluator comments were incorporated.	Fujii	Ito
6	March 8, 2007	1.05	Evaluator comments were incorporated.	Fujii	Ito
7	March 27, 2007	1.06	Evaluator comments were incorporated.	Fujii	Ito
8	April 23, 2007	1.07	Evaluator comments were incorporated.	Fujii	Ito
9	May 10, 2007	1.08	Evaluator comments were incorporated.	Fujii	Ito

Trademarks

- Linux is a registered trademark of Linus Torvalds.
- Microsoft is a registered trademark of Microsoft Corp. in the U.S. and other countries.
- Red Hat is a trademark or a registered trademark of Red Hat Inc. in the United States and other countries.
- Solaris is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.
- All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc., in the United States and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.
- Sun is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.
- Sun Microsystems is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.
- Windows is a registered trademark of Microsoft Corp. in the U.S. and other countries.
- Windows Server is a registered trademark of Microsoft Corp. in the U.S. and other countries.
- Microsoft Internet Explorer is a product name of Microsoft Corporation in the U.S. and other countries.
- Mozilla is a trademark of the Mozilla Foundation in the U.S and other countries.

Copyrights

HiCommand Suite Common Component Security Target

- Contents -

1. ST Introduction 5
1.1. ST identification 5
1.1.1. ST identification information 5
1.1.2. TOE identification information 5
1.1.3. Applicable CC 5
1.2. ST overview 6
1.3. CC conformance 6
1.4. Definition of terms 7
2. TOE Description 8
2.1. TOE type 8
2.2. System with the TOE and TOE overview 8
2.2.1. Overview of a system with the TOE 8
2.3. How the TOE is used 10
2.3.1. TOE model 10
2.3.2. TOE users 11
2.3.3. Hardware configuration 11
2.3.4. Software configuration 13
2.4. TOE boundaries 14
2.4.1. Physical TOE boundary 14
2.4.2. Logical TOE boundary 15
2.5. Assets 18
3. TOE Security Environment 19
3.1. Assumptions 19
3.2. Threats 20
3.2.1. Threat agents 20
3.2.2. Identifying threats 20
3.3. Organizational security policies 20
4. Security Objectives 21
4.1. Security Objectives for the TOE 21
4.2. Security Objectives for the environment 21
4.2.1. Security Objectives for the IT environment 21
4.2.2. Security Objectives achieved during operations 21
5. IT Security Requirements 23

5.1. TOE security requirements 23
5.1.1. TOE security functional requirements 23
5.1.2. Minimum function strength level 32
5.1.3. TOE security assurance requirements 32
5.2. Security requirements for the IT environment 32
5.2.1. Security functional requirements for the IT environment 33
6. TOE Summary Specification 34
6.1. TOE security functions 34
6.1.1. Identification and authentication function (**SF.I&A**) 34
6.1.2. Security information management function (**SF.MGMT**) 36
6.1.3. Warning banner function (**SF.BANNER**) 38
6.2. Strength of security functions 39
6.3. Assurance Measures 40
7. Protection Profile (PP) Claims 41
8. Rationale 42
8.1. Security Objectives Rationale 42
8.2. Security Requirements Rationale 45
8.2.1. TOE Security Functional Requirements Rationale 45
8.2.2. Rationale for the minimum strength of function level 47
8.2.3. Dependencies of security functional requirements 47
8.2.4. Dependencies of security assurance requirements 48
8.2.5. Internal consistency and mutual support of security functional requirements 48
8.2.6. Rationale for audit 49
8.2.7. Rationale for management requirements 49
8.2.8. Rationale for security assurance requirements 50
8.3. TOE Summary Specification Rationale 50
8.3.1. TOE security functions Rationale 50
8.3.2. Strength of Functions Rationale 54
8.3.3. Assurance Measures Rationale 54
8.4. PP Claims Rationale 54

1. ST Introduction

This section contains the ST identification, ST overview, CC conformance claim, and a definition of terms.

1.1. ST identification

1.1.1. ST identification information

The identification information of this security target (ST) is as follows:

ST title: HiCommand Suite Common Component Security Target
ST version: 1.08
Identification name: HSCC-ST-1.08
Date: May 10, 2007
Author: Hitachi, Ltd.

1.1.2. TOE identification information

The name of the product containing the target of evaluation (TOE) to be evaluated by this ST is as follows:

TOE name: HiCommand Suite Common Component
TOE version: 05-51-01
Developer: Hitachi, Ltd.
Applicable platforms:

- Platform running Java™VM (Version 1.4.2_03) that has been installed by HiCommand Suite Common Component for Windows
- Platform running Java™VM (Version 1.4.2_03) that has been installed by HiCommand Suite Common Component for Solaris
- Platform running Java™VM (Version 1.4.2_03) that has been installed by HiCommand Suite Common Component for Linux

Keyword: Storage management software

1.1.3. Applicable CC

This ST conforms to the following CC:

CC version 2.3

1.2. ST overview

The target of evaluation, HiCommand Suite Common Component (abbreviated hereafter to *HSCC*), runs as the base module that provides the common functions for storage management software that centrally manages multiple storage devices connected in a SAN environment.

The storage management software includes HiCommand Device Manager (abbreviated hereafter to *HDvM*), HiCommand Replication Monitor (abbreviated hereafter to *HRpM*), and HiCommand Tiered Storage Manager (abbreviated hereafter to *HTSM*), etc. These products and HSCC are generically referred to as HiCommand Suite.

HSCC is bundled with each product package as the base module of HiCommand Suite.

The HSCC security functions are as follows:

- Identification and authentication functions
- Security information management functions
- Warning banner functions

1.3. CC conformance

This ST claims conformance to the following CC:

- CC version 2.3 Part 2
- CC version 2.3 Part 3

This ST complies with evaluation assurance level EAL2 augmented with ALC_FLR.1 (Basic Flaw Remediation) .

This ST does not conform to any protection profiles (PP) .

1.4. Definition of terms

Table 1 briefly defines the terms and abbreviations used in this ST.

Table 1: Meaning of terms and abbreviations

Term	Meaning
ACL	Abbreviation for Access Control List.
SAN	Abbreviation for Storage Area Network.
Token	Identifier used by HSCC for managing sessions.
HSCC	HiCommand Suite Common Component. Part of HiCommand Suite. HSCC is the base module that provides common functions for the storage management software contained in HiCommand Suite.
HDvM	HiCommand Device Manager. Storage management software that is part of HiCommand Suite. HDvM provides volume management functionality for storage systems.
HRpM	HiCommand Replication Monitor. Storage management software that is part of HiCommand Suite. HRpM provides functionality for monitoring the copies that are made between the volumes in a storage system.
HTSM	HiCommand Tiered Storage Manager. Storage management software that is part of HiCommand Suite. HTSM controls the movement of data between the volumes in a storage system.
Security parameter	Parameter associated with the HSCC security functions.
Warning banner	Warning text to be displayed before users use the storage management software. A warning banner is mainly used to call attention to illegal use.

2. TOE Description

2.1. TOE type

This TOE is a software product that operates as the base module that provides the common functions for the storage management software contained in HiCommand Suite.

2.2. System with the TOE and TOE overview

2.2.1. Overview of a system with the TOE

A storage system (abbreviated hereafter to *storage*) contains multiple volumes in a frame, is connected to an application server that runs business applications, and stores information required to run business applications. As the size of an information system increases, so too does the size of the storage. When an information system is used, its storage must be managed. This means that the following tasks must be performed in satisfactory manner:

- Allocation of volumes (HDvM enables access from an application server).
- Monitoring of copies (HRpM enables the copying of the volumes that contain business data to be monitored).
- Movement of data (HTSM enables the movement of old data to another storage to create free volumes).

The storage administrator uses the management device connected to the target storages to centrally execute the above tasks on many volumes and storages by using storage management software that has the most appropriate functions. HiCommand Suite provides a group of software products that perform the storage management described above. Figure 1 shows an overview of a system that uses HiCommand Suite to perform storage management.

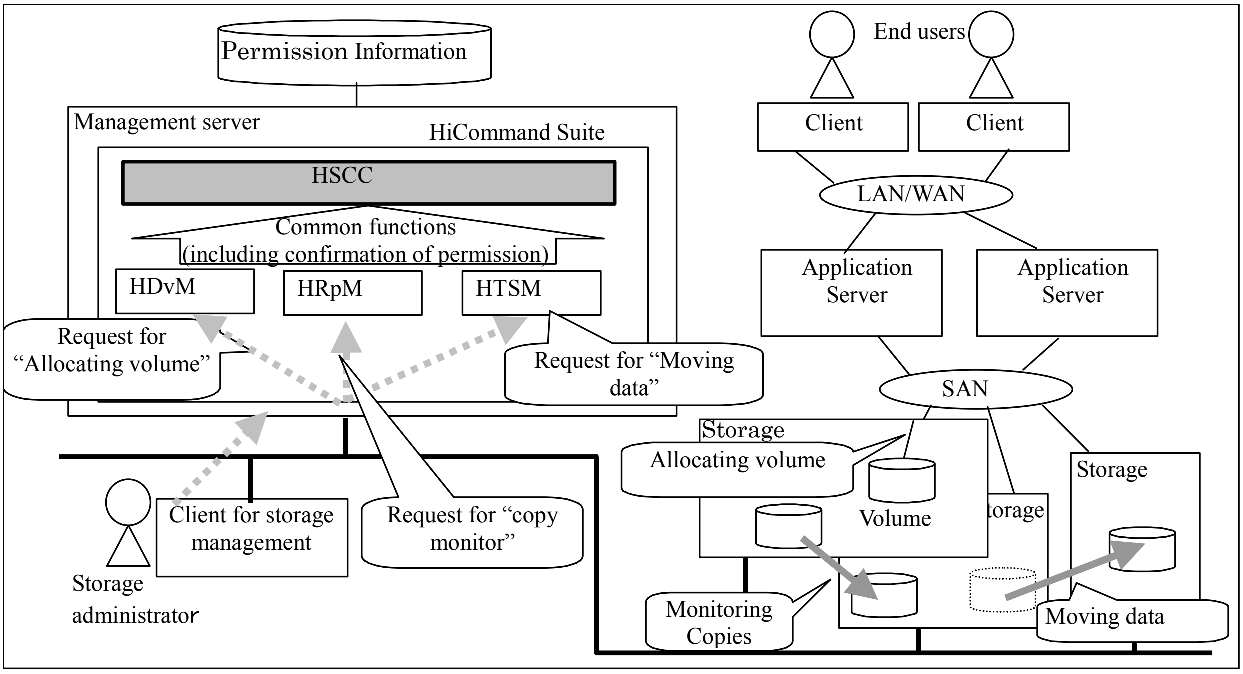

Figure 1: Overview

In Figure 1, the storage administrator accesses storage management software on a client terminal to perform copying or other operations and to request required operations. The TOE provides the common functions of the storage management software such as authentication, the display of permissions, and a graphical user interface for displaying information on the client terminal used to manage storages.

To enable the storage management requests from the storage administrator, such as allocating volumes and monitoring copies, to be performed within the authorized scope, the TOE authenticates the storage administrator and controls access to permission information before the requests for storage management are executed.

The TOE also provides account information for authentication as well as functions that allow account administrators to set permission information.

2.3. How the TOE is used

2.3.1. TOE model

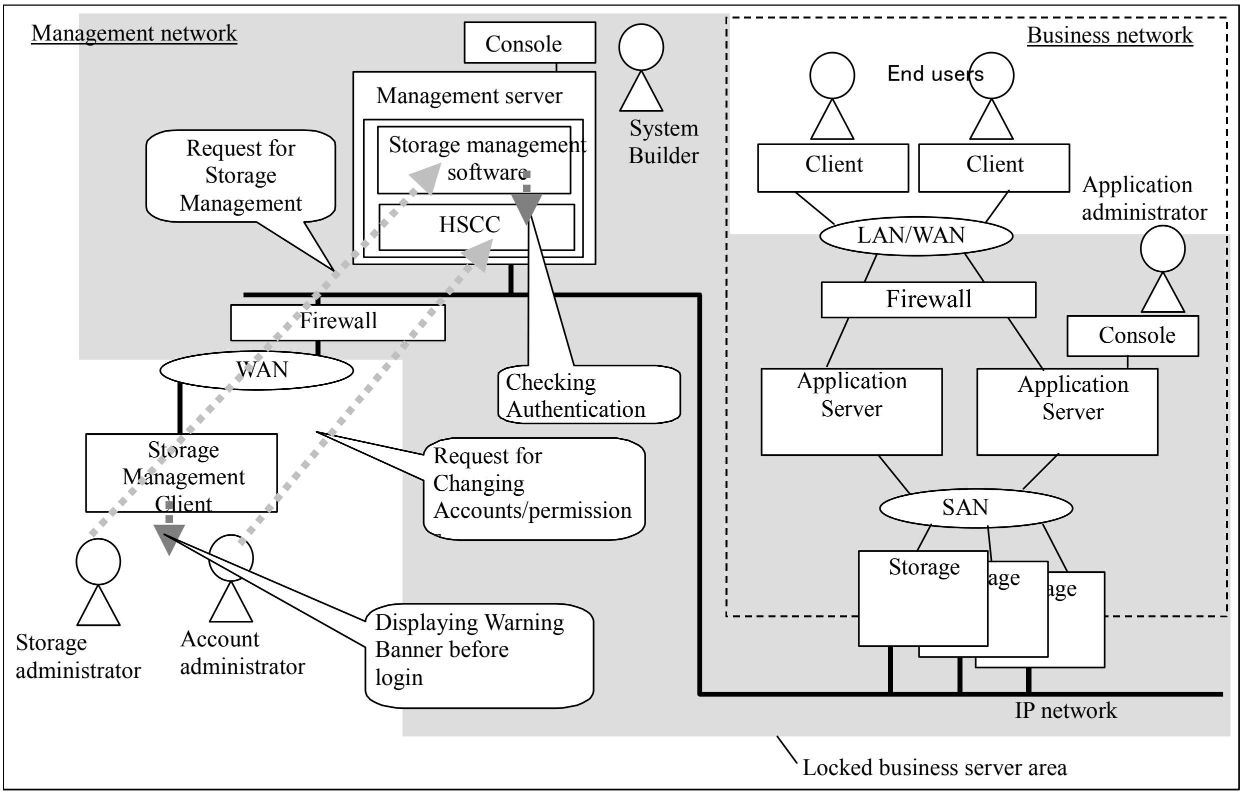

Figure 2: TOE model

In Figure 2, solid lines indicate physical cabling and devices, and dotted lines indicate either actions by user and software, or logical boundary of network. The shaded portion indicates the locked business server area, such as a computer center.

The business server area holds management servers, application servers, storages, and peripheral devices. Physical entry to this area is restricted.

Connected devices in the management network are management servers, storages, and peripheral devices. Connected devices in the business network are application servers, storages, and peripheral devices. Each of these networks is protected by a firewall from external access. The management network and the business network within these firewalls are generically called the internal network. The networks outside these firewalls are called the external network.

Each storage that belongs to both networks has two independent NICs, one that connects to the management network and the other that connects to the business network. The use of two NICs ensures separation of the management network and the business network, preventing mutual interference.

The storage administrator and the account administrator use the client terminal for storage management to access the TOE from an external network to issue requests to the storage management software for the performance of operations. When the administrators log on, warning banners are displayed to draw attention to any illegal use. In addition, users are instructed to use passwords that are difficult to guess.

2.3.2. TOE users

This ST assumes the following types of users. Users perform operations within a predefined scope of permissions.

(1) System builder (server / network administrator)

Role: Maintains and manages the system by, for example, backing up server data.

Permissions: Allowed to determine and set parameters required for building and running the system. Accordingly, the system builder can update (change and delete) the permissions of users, which are user data. The system builder's permissions are not changed.

Level of trust: Has responsibility for the system and is trusted.

(2) Account administrator

Role: Manages the accounts of users who use the system and specify settings for the system.

Permissions: Allowed to manage accounts based on the source information for an account. The source information for an account includes determining whether an account should be created and which permissions should be granted to the account, and is derived from organizational information such as the organizational hierarchy. Accordingly, the account administrator can update (change and delete) the permissions of users, which are user data.

Level of trust: Has responsibility for own work and is trusted within the scope of that work.

(3) Storage administrator

Role: Manages storages by, for example, managing the resources in the storages.

Permissions: Allowed to allocate resources in the storages installed by the system builder. Accordingly, the storage administrator can access the permissions of users, which are user data, to determine the permissions granted to the storage administrator.

Level of trust: Has responsibility for own work and is trusted within the scope of that work.

2.3.3. Hardware configuration

This subsection describes the hardware requirements for running the TOE. Hardware requirements are provided for each of the platforms described in Section 1.1.

(１) For Windows

Devices in the following series that support the Windows platform described in Section 1.1:

- Hitachi FLORA series
- Hitachi HA8000 series
- Non-Hitachi PC/AT-compatible devices
- Hitachi BladeSymphony series

The minimum requirements are as follows:

CPU clock:	1 GHz
Memory size:	512 MB
Disk size:	4 GB

(２) For Linux

Devices in the following series that support the Linux platform described in Section 1.1:

- Hitachi FLORA series
- Hitachi HA8000 series
- Non-Hitachi PC/AT-compatible devices
- Hitachi BladeSymphony series

The minimum requirements are as follows:

CPU clock:	1 GHz
Memory size:	1 GB
Disk size:	4 GB

(３) For Solaris

Devices in the following series that support the Solaris platform described in Section 1.1:

- Solaris SPARC

The minimum requirements are as follows:

CPU clock:	1 GHz
Memory size:	1 GB
Disk size:	4 GB

2.3.4. Software configuration

This subsection describes the software requirements for running the TOE.

(４) For Windows

- Applicable platform described in Section 1.1
- Microsoft Internet Explorer browser

(５) For Linux

- Applicable platform described in Section 1.1
- Mozilla browser

(６) For Solaris

- Applicable platform described in Section 1.1
- Mozilla browser

2.4. TOE boundaries

2.4.1. Physical TOE boundary

The physical TOE boundary is defined by the following libraries and programs.

Figure 3 shows the software configuration that includes the TOE. The TOE is HSCC. The modules implementing the TOE security functions are shaded.

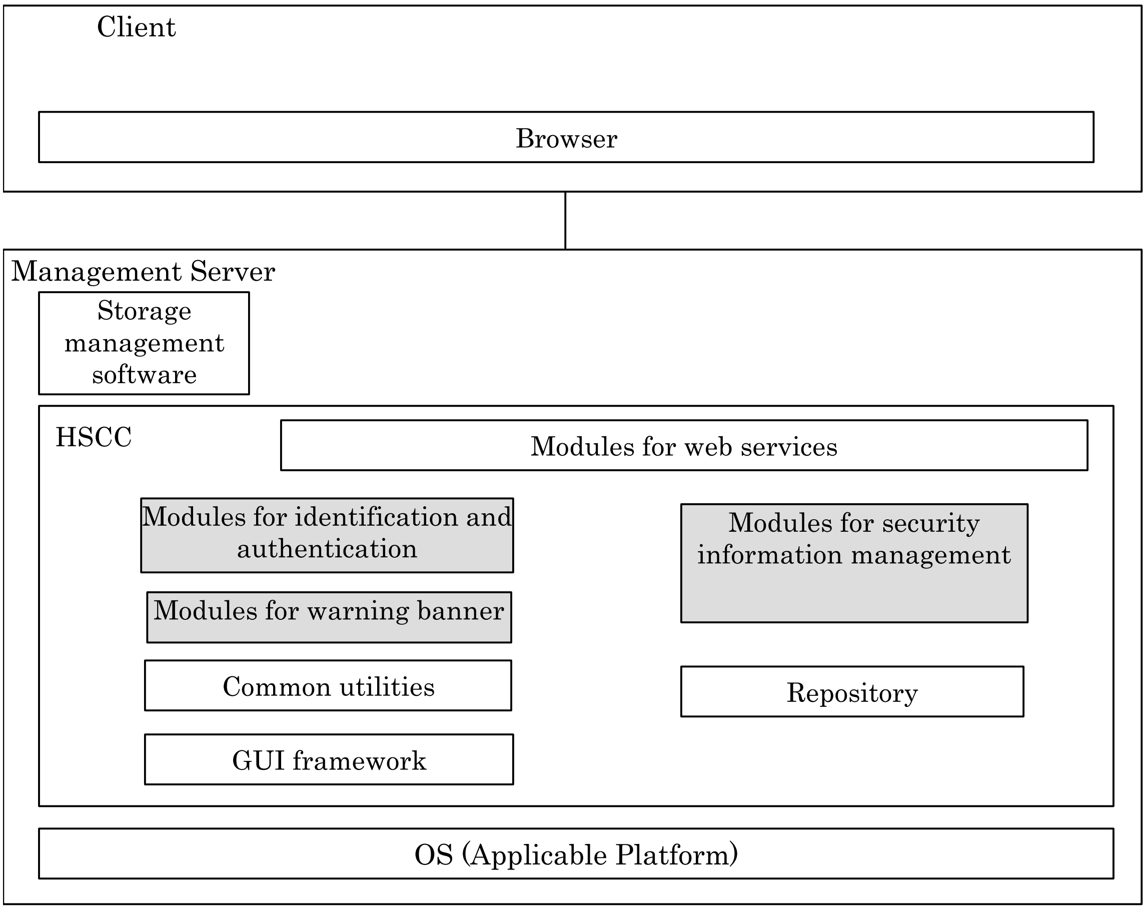
Client
Browser
Management Server
Storage management software
HSCC
Modules for web services
Modules for identification and authentication
Modules for security information management
Modules for warning banner
Common utilities
Repository
GUI framework
OS (Applicable Platform)

Figure 3: Software configuration including the TOE

Identification and Authentication Module is a module that implements the identification and authentication function of the TOE.

Security Information Management Module is a module that implements the security information management function of the TOE.

Warning Banner Module is a module that implements the warning banner functionality of the TOE.

Common Utility is a module that implements the common functions of the TOE.

Web Service Module is a module that implements the TOE Web service.

GUI Framework is a module that implements the TOE graphical user interface.

Repository is the database that stores data for the TOE.

2.4.2. Logical TOE boundary

Table 2 lists the TOE functions. The TOE security functions are shaded.

Table 2: Functions of the TOE (HSCC)

Function	Overview
Identification and Authentication	Uses user IDs and passwords to authenticate users as the basis for maintaining sessions. In addition, based on the authentication, the function passes a permission to the requesting user.
Security Information Management	Manages account information, permissions, and banner information (creation, viewing, change, and deletion of banner information). This function also sets security parameters.
Warning Banner	Provides warning messages for HiCommand Suite.
Common Utility	Used for setting up and administering HiCommand Suite.
Web Service	Provides a Web service so that HiCommand Suite can interact with the browsers on client terminals.
GUI Framework	Provides a GUI framework for HiCommand Suite.
Repository	Memory area for storing the data used to run HiCommand Suite.

(1) Identification and Authentication function

This function identifies a TOE user when the user logs on to the storage management software, and passes a permission to the user when the user is authenticated. A permission defines which storage management software the user is allowed to use. For example, a user with Modify permission can set and change resources managed by the storage management software. A user with View permission can only view such resources.

If successive authentication attempts by the user fail for a predefined number of times during the identification and authentication period, the user's account for the TOE is automatically locked.

(2) Security Information Management function

This function manages the user IDs, passwords, and lock status of TOE users as account information. When a password is set, the function checks whether the password satisfies the conditions set in the security parameters. In addition, when a permission is entered for a corresponding user ID, the function stores the permission in the ACL.

The function stores the variable parameters for automatic account locking and password complexity checking as security parameters.

The function manages the warning messages for illegal use of storage management software as banner information, and provides methods that allow TOE users to create, delete, and change banners upon request.

(3) Warning Banner function

This function returns banner information in response to a request from the storage management software.

The TOE protects the ACL that stores permissions from changes by users who do not have the appropriate permissions when the TOE sends permissions to the storage management software. The ACL is associated with user IDs and has the security attributes of TOE user roles (such as the account administrator role). The ACL contains permissions that enable data to be referenced and changed via the storage management software. When the TOE identifies and authenticates users, it reads security attributes for the corresponding user IDs as needed and uses the attributes as access permission information (session data).

The TOE also provides functions that allow the account administrators to set account information for authenticating users and to set security attributes. Once the account administrator has been identified and authenticated by the TOE, the account administrator can access the security information management function of the TOE from a client terminal and perform account management, such as creating, updating, and deleting user accounts, and setting permissions. Generally, an ACL is TSF data used for access control and is managed by special administrators. However, for the security functions claimed by this TOE, the permissions in the ACL are treated as user data. The TOE permits access, such as viewing and updating by a user such as the storage administrator, to the information in the ACL based on the role of the user.

The following explains how to use the TOE.

(1) Preparation by the system builder

- The system builder purchases required information system resources, including the TOE.
- The system builder installs and connects the devices on which the TOE is to be installed, builds the prerequisite environment for the TOE, installs the TOE, performs setup, and verifies correct operation.
- The system builder creates an account for the account administrator with the appropriate account management permission based on the default account and default password, and notifies the account administrator of this information.

(2) Account management by the account administrator

- The account administrator acquires an appropriate account and password.
- The account administrator uses the appropriate account and password to access the TOE to be authenticated by the TOE.
- The account administrator creates the accounts for other account administrators and storage administrators in the TOE based on the source information for the accounts to be set. The account administrator also sets attributes such as permissions for the created accounts.

- The account administrator notifies other account administrators and storage administrators of the created account information.

(3) Storage management by storage administrators

- The storage administrator acquires an appropriate account and password.
- The storage administrator uses the appropriate account and password to access the TOE to obtain authentication by the TOE. After authentication, the storage administrator acquires the permission corresponding to the account.
- After the authentication by the TOE, the storage administrator performs storage management to the extend allowed by the acquired permission.

2.5. Assets

Since the main purpose of the TOE is to allow storage administrators to acquire an authorized storage management environment by acquiring appropriate permissions through authentication, the following assets are protected by the TOE:

- Permissions

  Permissions are granted to accounts and stored in the ACL together with corresponding user IDs and security attributes.

- Banner information

  Text used by the warning banner function.

3. TOE Security Environment

This section describes assumptions, threats, and organizational security policies.

3.1. Assumptions

A.PHYSICAL (management of hardware)

The management server on which the TOE and storage management software run, peripheral devices, storage devices, the internal network, and firewall at the boundary of the internal network are installed in the physically isolated business server area. Only authorized administrators are permitted to enter this area.

A.NETWORKS (networks)

The internal network containing the management network connected to the management server is installed in the business server area and performs only the communication that is necessary. A firewall that monitors traffic logically separates the internal network from external networks and detects traffic that is inappropriate.

A.ADMINISTRATORS (administrators)

The system builder is trusted. Account administrators, storage administrators, and administrators of other servers, including application servers, do not perform malicious acts with regard to one another's work. Work includes the management of accounts and permissions of storage management software users, the management of storages, and the management of other servers.

A.SECURE_CHANNEL (communication secrecy)

The network between the management server on which the TOE and storage management software run, and the management clients is secure with regard to secrecy and completeness of communication.

A.TOKEN (available tokens)

The TOE does not create an environment containing products with either tokens that are generated outside the TOE or tokens of insufficient strength.

A.PASSWORD (complex passwords)

Authentication methods have sufficient strength so that illegal users cannot log on to the system by guessing passwords.

3.2. Threats

3.2.1. Threat agents

A threat agent that intentionally or accidentally breaches security is defined as follows:

- Illegal user (user who is not authorized to use the TOE and all of the storage management software)
- Storage administrator (person who is authorized to use the TOE and one of the storage management software programs)

3.2.2. Identifying threats

T.ILLEGAL_ACCESS (illegal connection)

From a management client, an illegal user might delete, change, or expose the permissions managed by the TOE for the storage management software functions, or delete or change banner information.

T.UNAUTHORISED_ACCESS (unauthorized access)

From a management client, an authenticated storage administrator or account administrator might perform an unauthorized operation that deletes, changes, or exposes the permissions managed by the TOE, or might delete or change banner information.

3.3. Organizational security policies

P.BANNER (warning banners)

Storage management software must have functions that display advisory warning messages related to its illegal use of the software.

4. Security Objectives

This section describes the security objectives for the TOE and the security objectives for the environment.

4.1. Security Objectives for the TOE

O.I&A

The TOE shall identify and authenticate users so that only authorized users can access the permissions managed by the TOE for the storage management software functions.

O.MGMT

The TOE shall provide methods for viewing and setting permissions, roles, and banner information, and control access so that users with the appropriate permissions can use the methods.

O.BANNER

The TOE shall provide storage management software with advisory warning messages regarding illegal use of the storage management software.

O.PASSWORD

The TOE shall limit the patterns for passwords that can be registered for user accounts for the storage management software based on the values of preset security parameters.

4.2. Security Objectives for the environment

4.2.1. Security Objectives for the IT environment

OE.SECURE_CHANNEL

The network between the management server and management clients shall use protected communication paths based on encryption or other methods to protect against exposure and changes.

OE.BANNER

The storage management software shall have functionality that displays advisory messages (provided by the TOE) regarding illegal use.

4.2.2. Security Objectives achieved during operations

OM.PHYSICAL

The management server on which the TOE and storage management software run, peripheral devices, storage devices, the internal network, and the firewall at the boundary of the internal network shall be installed in the physically isolated business server area. Only authorized administrators are permitted to enter and leave this area.

OM.FIREWALL

A firewall shall be installed between the internal network (contains the management network connected to the management server) installed in the business server area and the external network. The firewall shall be set up and shall monitor to detect illegal traffic so that unnecessary communication from the external network does not enter the networks within the business server area.

OM.ADMINISTRATORS

The head of the organization shall select appropriate personnel to guarantee that the system builder can be trusted and that account administrators, storage administrators, and administrators of other servers, including application servers, shall not perform malicious acts with regard to one another's work. Work includes the management of the accounts and permissions of storage management software users, the management of storages, and the management of other servers.

OM.TOE_ACCOUNT

The system builder, account administrators, and storage administrators must not expose the passwords that they create for the user accounts that use the storage management software. Passwords shall be made difficult to guess and shall be changed at an appropriate frequency.

OM.TOKEN

The system builder shall not build an environment containing the TOE and products with the following tokens:

- Tokens generated by an entity other than the TOE
- Tokens from which user IDs and passwords can be guessed

OM.PASSWORD

The system builder and account administrators shall specify settings that require complex passwords and shall limit the number of repeated authentication attempts to prevent guessing by illegal users of passwords at logon.

5. IT Security Requirements

5.1. TOE security requirements

This section describes the TOE security requirements. All the functional requirement components consist of the components that are defined in CC Part 2.

5.1.1. TOE security functional requirements

FDP_ACC.1 Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the [assignment: access control SFP] on [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP].

Dependencies: FDP_ACF.1 Security attribute based access control

[assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

Subject:	Process acting on behalf of the user
Object:	ACL table, banner information file
Operation:	Viewing, change, creation, or deletion

[assignment: access control SFP]

ACL access control SFP

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the [assignment: access control SFP] to objects based on the following: [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects].

FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects].

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

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

[assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes] and [assignment: access control SFP]

List of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes	Access control SFP
Subject: Process acting on behalf of the user Object: ACL table Subject attributes: User ID and role associated with the subject Object attribute: User ID of the object	ACL access control SFP
Subject: Process acting on behalf of the user Object: Banner information file Subject attributes: User ID and role associated with the subject Object attribute: None	ACL access control SFP

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

Subject	Object	Rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects
Process acting on behalf of the user	ACL table	Only when the user ID associated with the subject matches a user ID of the object, the process can refer the user's role and permission.
Process acting on behalf of the user	ACL table	When the user ID associated with the subject matches a user ID of the object and the user’s role is account administrator or system builder, the process can create, delete, and change the user’s roles and permissions.
Process acting on behalf of the user	Banner information file	When the role associated with the subject is account administrator or system builder, the process can create, delete, and change banner information.

[assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]

Subject	Object	Rules, based on security attributes, that explicitly authorise access of subjects to objects
Process acting on behalf of the user	Banner information file	Viewing of banner information is always authorised.

[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

Subject	Object	Rules, based on security attributes, that explicitly deny access of subjects to objects
Process acting on behalf of the user	ACL table	Even if the user ID associated with the subject matches a user ID of the object and the role is account administrator, the process cannot delete or change the identified user's role and permission.
Process acting on behalf of the user	ACL table	When the object provides the system builder role and the corresponding permission, the process cannot delete or change the role and permission.

FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the [assignment: access control SFP, information flow control SFP] to restrict the ability to [selection: change_default, query, modify, delete, [assignment: other operations]] the security attributes [assignment: list of security attributes] to [assignment: the authorised identified roles].

Dependencies: [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

The following table lists the assignments and selections described above.

Security attribute	Selection: Change_default, query, modify, delete Assignment: Other operations	Authorised identified roles	Access control SFP, information flow control SFP
User ID and role associated with the object other than the user IDs of the system builder and the subject	Selection: Modify, delete Assignment: None	Account administrator, system builder	ACL access control SFP
User ID and role associated with the object, which are the same as the user ID of the system builder or the subject	Selection: None Assignment: None	-	ACL access control SFP

FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the [assignment: access control SFP, information flow control SFP] to provide [selection, choose one of: restrictive, permissive, [assignment: other property]] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [assignment: the authorised identified roles] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

[selection, choose one of: restrictive, permissive, [assignment: other property]]
Select "restrictive".

[assignment : other property]
None

[assignment: access control SFP, information flow control SFP]
ACL access control SFP

[assignment: the authorised identified roles]
None

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to [selection: change_default, query, modify, delete, clear, [assignment: other operations]] the [assignment: list of TSF data] to [assignment: the authorised identified roles].

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

The following table lists the assignments and selections described above.

TSF data	Selection: Change_default, query, modify, delete, clear Assignment: Other operations	Authorised identified roles
User ID other than the system builder	Selection: Delete Assignment: Register	Account administrator, system builder
User ID of the system builder	Selection: None Assignment: None	-

Password associated with the user ID	Selection: Modify, delete Assignment: Register	Account administrator, system builder
	Selection: Modify	Storage administrator corresponding to the user ID
Lock status	Selection: Query, modify	Account administrator, system builder
Security parameter	Selection: Query, modify, clear Assignment: None	Account administrator, system builder

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 management functions: [assignment: list of management functions to be provided by the TSF].

Dependencies: No dependencies.

The following table lists the assignments described above.

Table 3: Security management functions provided by the TSF

Functional requirement	Management requirement	Management item
FDP_ACC.1	None	None
FDP_ACF.1	a) Managing the attributes used to make explicit access or denial based decisions.	a) Management of user IDs and associated permissions
FMT_MSA.1	a) Managing the group of roles that can interact with the security attributes.	a) None (no groups of roles that may affect security attributes that may affect roles exist)
FMT_MSA.3	a) Managing the group of roles that can specify initial values; b) Managing the permissive or restrictive setting of default values for a given access control SFP.	a) None (no groups of roles exist) b) None (no management of default value settings exists)
FMT_MTD.1	a) Managing the group of roles that can interact with the TSF data.	a) None (no groups of roles that may affect TSF data that may affect roles exist)
FMT_SMR.1	a) Managing the group of users that are part of a role.	a) None (no groups of users that consist of parts of roles exist)

FIA_UAU.1	a) Management of the authentication data by an administrator; b) Management of the authentication data by the associated user; c) Managing the list of actions that can be taken before the user is authenticated.	a) Creation and change of passwords b) Change of passwords by users c) None (lists are not changed)
FIA_UID.1	a) The management of the user identities; b) If an authorised administrator can change the actions allowed before identification, the managing of the action lists.	a) Creation and deletion of user IDs for accounts b) None (lists are not changed)
FIA_SOS.1	a) The management of the metric used to verify the secrets.	a) Specification of the required number of characters and types of characters in passwords when passwords are set
FIA_ATD.1	a) If so indicated in the assignment, the authorised administrator might be able to define additional security attributes for users.	a) None (no additional security attributes are defined)
FIA_USB.1	a) An authorised administrator can define default subject security attributes. b) An authorised administrator can change subject security attributes.	a) None (no security attributes are given by default) b) None (since no security attributes are given by default)
FIA_AFL.1	a) Management of the threshold for unsuccessful authentication attempts; b) Management of actions to be taken in the event of an authentication failure.	a) Setting and changing of threshold values by administrators b) None (the only action to be performed is locking accounts)
FTA_TAB.1	a) Maintenance of the banner by the authorised administrator.	a) Setting of banner contents by administrators
FPT_RVM.1	None	None
FPT_SEP.1	None	None

FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles [assignment: the authorised identified roles].

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

Dependencies: FIA_UID.1 Timing of identification

[assignment: the authorised identified roles]
Storage administrator, account administrator, system builder

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

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

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

Dependencies: FIA_UID.1 Timing of identification

[assignment: list of TSF mediated actions]
Warning banner function

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

FIA_UID.1.1 The TSF shall allow [assignment: list of TSF-mediated actions] on behalf of the user to be performed before the user is identified.

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

Dependencies: No dependencies.

[assignment: list of TSF-mediated actions]
Warning banner function

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 [assignment: a defined quality metric].

Dependencies: No dependencies.

[assignment: a defined quality metric]

Password generation condition written in a security parameter

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

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

Dependencies: No dependencies.

[assignment: list of security attributes]

User ID, role

FIA_USB.1 User-subject binding

Hierarchical to: No other components.

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

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: [assignment: rules for the initial association of attributes].

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: [assignment: rules for the changing of attributes].

Dependencies: FIA_ATD.1 User attribute definition

The following table lists the assignments and selections described above.

User security attribute	Rules for the initial association of attributes	Rules for the changing of attributes
User ID and role associated with the object	None	None

FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

FIA_AFL.1.1 The TSF shall detect when [selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]] unsuccessful authentication attempts occur related to [assignment: list of authentication events].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [assignment: list of actions].

Dependencies: FIA_UAU.1 Timing of authentication

[assignment: list of authentication events]
Authenticated account of the user after the last successful authentication (except for the system builder)

[selection: [assignment: positive integer number], an administrator configurable positive integer within [assignment: range of acceptable values]]
Selection: An administrator configurable positive integer within [assignment: range of acceptable values]
Range of acceptable values: Range of values specified in security parameters

[assignment: list of actions]
Lock an account (except for the system builder).

FTA_TAB.1 Default TOE access banners

Hierarchical to: No other components.

FTA_TAB.1.1 Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorised use of the TOE.

Dependencies: No dependencies.

FPT_RVM.1 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 the TSF 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.

5.1.2. Minimum function strength level

The claimed minimum strength of function of this TOE is SOF-basic. The security functional requirements based on a probabilistic or permutational mechanism are FIA_UAU.1 and FIA_SOS.1.

5.1.3. TOE security assurance requirements

The evaluation assurance level of this TOE is EAL2 augmented with ALC_FLR.1.

All the assurance requirement components are directly derived from the assurance components specified in CC Part 3. Table 4 lists the assurance components added by EAL2 augmented (EAL2+ALC_FLR.1).

Table 4: Assurance components - EAL2 augmented (EAL2+ALC_FLR.1)

Assurance class	Assurance component	
Configuration management (ACM class)	**ACM_CAP.2**	Configuration items
Delivery and operation (ADO class)	**ADO_DEL.1**	Delivery procedures
	ADO_IGS.1	Installation, generation, and start-up procedures
Development (ADV class)	**ADV_FSP.1**	Informal functional specification
	ADV_HLD.1	Descriptive high-level design
	ADV_RCR.1	Informal correspondence demonstration
Guidance documents (AGD class)	**AGD_ADM.1**	Administrator guidance
	AGD_USR.1	User guidance
Life cycle support (ALC class)	**ALC_FLR.1**	Basic flaw remediation
Tests (ATE class)	**ATE_COV.1**	Evidence of coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability assessment (AVA class)	**AVA_SOF.1**	Strength of TOE security function evaluation
	AVA_VLA.1	Developer vulnerability analysis

5.2. Security requirements for the IT environment

This section describes the functional requirements in the security functions provided by the IT environment. All the functional requirement components derive from the components specified in CC

Part 2.

5.2.1. Security functional requirements for the IT environment

FPT_ITC.1 Inter-TSF confidentiality during transmission

Hierarchical to: No other components.

FPT_ITC.1.1 [refinement: network between the management server and management clients] shall protect all TSF data transmitted from the TSF to a remote trusted IT product from unauthorised disclosure during transmission.

Dependencies: No dependencies.

FTA_TAB.1E Default TOE access banners

Hierarchical to: No other components.

FTA_TAB.1.1 Before establishing a user session, [refinement: storage management software] shall display an advisory warning message regarding unauthorised use of the TOE.

Dependencies: No dependencies.

6. TOE Summary Specification

This section describes the security functions, the strength of the security functions, and the security assurance measures of the TOE.

6.1. TOE security functions

This section describes the security functions of the TOE. As shown in Table 5, the security functions described in this section satisfy the TOE security functional requirements described in Subsection 5.1.1.

Table 5: Correspondence between TOE security functions and TOE security functional requirements

TOE security functional requirement TOE security function	FDP_ACC.1	FDP_ACF.1	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FIA_UAU.1	FIA_UID.1	FIA_SOS.1	FIA_ATD.1	FIA_USB.1	FIA_AFL.1	FIA_TAB.1	FPT_RVM.1	FPT_SEP.1
SF.I&A	X	X						X	X		X	X	X		X	X
SF.MGMT	X	X	X	X	X	X	X			X					X	X
SF.BANNER	X	X												X	X	

6.1.1. Identification and authentication function (**SF.I&A**)

When a user uses the storage management software and the TOE, **SF.I&A** identifies and authenticates the user. In response to a request from the storage management software, SF.I&A manages the session of the user that has logged on and checks whether the identification and authentication of that user are maintained.

(1) Identifying and authenticating

When a user logs on to the storage management software or when the storage management software executes the security information management function provided by **SF.MGMT**, **SF.I&A** accepts an identification and authentication request based on the user ID and password for the user account, compares the user account with the registered account information (user ID, password, lock status (locked or unlocked) of the user), and identifies and authenticates the user account.

When identification and authentication of the user account are successful and the user account is unlocked, **SF.I&A** accesses the ACL table to obtain the role and permission of the user. At this time, **SF.I&A** controls access to the ACL table (object) based on the user ID associated with the process (subject) acting on behalf of the user and the following rule:

- Only when the user ID associated with the subject matches a user ID in the object can the user have access to the user's permission.

When the role and permission acquired as described above allow the target storage management software to be used, **SF.I&A** proceeds to the session management processing described below.

When **SF.I&A** is unable to identify or authenticate a user, or when the user account is locked, or when the acquired role and permission do not allow the use of the target storage management software, **SF.I&A** returns an error to the storage management software.

Until **SF.I&A** is able to identify and authenticate a user, no action is executed except for outputting a warning message provided by the warning banner function (**SF.BANNER**).

SF.I&A guarantees that the processing described is always performed when it receives a request from the storage management software to identify and authenticate a user.

SF.I&A guarantees that the access control described above is always performed when the above process acting on behalf of the user accesses the ACL table.

(2) Automatically locking accounts

When **SF.I&A** identifies and authenticates users that log on to the storage management software, **SF.I&A** automatically locks a user account whose repeated authentication attempts fail for the preset number of times. The exception is the account of the system builder, which **SF.I&A** does not lock. When locked, accounts are locked indefinitely. **SF.MGMT** unlocks user accounts and sets the threshold value for the number of repeated failures that is applied to automatically lock the accounts whose authentication fails. **SF.I&A** manages the number of repeated authentication attempt failures for each user account. Only when authentication is successful, and when the preset number of repeated authentication attempt failures reaches the threshold value and an account is locked, does **SF.I&A** clear the number of repeated failures for the account. When an account is automatically locked, if another session with the same account has already logged on to the storage management software, the automatic locking of the account does not affect the operation of the successfully authenticated session.

(3) Managing sessions

When **SF.I&A** is able to identify and authenticate a user account as described above and the necessary role and permission are acquired, **SF.I&A** maintains and manages the user ID and role of the user as session data, and associates the user ID and role with the process that acts on behalf of the user.

When the storage management software requests execution of the security information management function provided by **SF.MGMT**, **SF.MGMT** proceeds to this processing. At this time, **SF.I&A** maintains and manages the session data described above while the security information management function is operating.

When the storage management software requests authentication of a user for logon, in response to the request, **SF.I&A** generates a token that identifies a user session for each logon. It then returns the user ID, role, permission, and token associated with the user that successfully logged on.

After a session for a user who has successfully logged on to the storage management software is established, **SF.I&A** checks the session data to confirm the validity of the user session. **SF.I&A** performs this processing if it receives a request from the storage management software or another TSF to check the validity of the user session that is using a token

If **SF.I&A** determines that the user session is valid, **SF.I&A** returns the user ID, role, and permission of the user in response to the request from storage management software.

If **SF.I&A** determines that the user session is not valid, **SF.I&A** returns an error to the storage management software or the other TSF.

SF.I&A does not change the role in the session data of a user if **SF.MGMT** has changed the role of the user in the ACL table while the user is logged on. Accordingly, the role that took effect at logon continues while the user is logged on to the storage management software.

When **SF.I&A** receives a logout request from a user, **SF.I&A** deletes the information related to the user session from the session data and ends the session.

Only authorized processes can access the user ID and role associated with each process that acts on behalf of a user that is successfully logged on. Therefore, **SF.I&A** guarantees that the user IDs and roles described above are not changed by untrusted processes, but are changed only by the processes that are successfully logged on and act on behalf of users.

6.1.2. Security information management function (**SF.MGMT**)

SF.MGMT manages account information, the ACL, banner information, and security parameters. Before **SF.MGMT** can be used, **SF.I&A** must be used beforehand to successfully identify and authenticate the user account.

(1) Managing accounts

SF.MGMT manages the correspondence among the user ID, password, and lock status (locked or unlocked) for each user account as account information. When a user sends a request, **SF.MGMT** permits operations for registering or deleting the user ID (account), for registering, changing, or deleting the password (the account itself is deleted), and for querying and changing the lock status.

SF.MGMT permits account administrators and the system builder to perform all of the above operations. For storage administrators, **SF.MGMT** only permits an administrator to change the administrator's own password. Note that **SF.MGMT** does not allow any user to register a new account that has the system builder role or to delete an account that has the system builder role.

(2) Checking the complexity of passwords

SF.MGMT checks whether a password satisfies the following quality criteria when a new account is created or a password is registered or changed. **SF.MGMT** does not allow a password that does not satisfy the quality criteria to be set.

- The number of characters in a password must meet a minimum number of characters, which is set in a security parameter.
- The types of characters that can be used in passwords must be alphabetic and numeric characters and symbols, and the password complexity conditions set in security parameters must be met.

(3) Managing the ACL

SF.MGMT manages the correspondence among the user ID, role, and permission for each user account as the ACL. In response to a request from a user, **SF.MGMT** accesses the ACL table and provides operations for registering, changing, and deleting roles and permissions.

SF.MGMT provides initial values for roles and permissions for user IDs when no role or permission has been set.

When a process acting on behalf of a user performs any of the above operations, **SF.MGMT** controls the access to the ACL table (object) based on the user ID and role associated with the process (subject) and the following rules:

- When the user ID associated with the subject matches a user ID in the object and the role is account administrator or system builder, the process can create, delete, and change the roles and permissions of users.
- Even when the user ID associated with the subject matches a user ID in the object and the role is account administrator, the process cannot delete or change the user's role and permission.
- When the object provides the system builder role and corresponding permission, the process cannot delete or change the role and permission.

SF.MGMT guarantees that the access control described above is always performed.

Only authorized processes can access the information in the ACL. Accordingly, **SF.MGMT** guarantees that only processes acting on behalf of the users that are successfully identified and authenticated, and not untrusted processes, can change the information in the ACL.

(4) Managing security parameters

SF.MGMT manages the variable parameters related to the TSF for automatic locking of accounts and complexity checking of passwords as security parameters. Table 6 lists the security parameters. In response to a request from a user, **SF.MGMT** provides operations for querying, modifying, and clearing the parameters.

SF.MGMT permits only account administrators and the system builder to perform these operations.

Table 6: Security parameters

#	Parameter	Description
1	Threshold value for the number of consecutive authentication attempt failures	Threshold value used by the automatic account lock function as the trigger for automatically locking accounts when repeated authentication attempts fail
2	Minimum number of characters in a password	Minimum number of characters in a password
3	Password complexity condition	Condition specifying that the specified number of the specified types of characters must be included in a password

(5) Managing banner information

SF.MGMT manages advisory warning messages regarding illegal use of storage management software as banner information. In response to a request from a user, **SF.MGMT** accesses the banner information file and provides operations for generating, deleting, and changing banner information.

When a process acting on behalf of a user performs these operations, **SF.MGMT** controls the access to the banner information file (object) based on the user ID and role associated with the process (subject) and the following rule:

- When the role associated with the subject is account administrator or system builder, banner information can be generated, deleted, or changed.

SF.MGMT guarantees that the access control described above is always performed.

Only the process that is authorized to use the banner information file editing function and the system builder, when successfully logged on to the management server, can access banner information. Accordingly, **SF.MGMT** guarantees that banner information is changed only by processes acting on the behalf of users who are successfully identified and authenticated, and not by untrusted processes.

6.1.3. Warning banner function (**SF.BANNER**)

SF.BANNER returns banner information that is set by **SF.MGMT** in response to a request from the storage management software. At this time, **SF.BANNER** controls access so that viewing of the banner information is always allowed. The banner information is the text of an advisory warning message regarding illegal use of the storage management software. The storage management software displays the warning message acquired as described in the logon window used for identifying and authenticating users.

SF.BANNER guarantees that the access control described above is always performed.

6.2. Strength of security functions

The security functions based on a probabilistic or permutational mechanism consist of the identification and authentication function (**SF.I&A**) for generating tokens and checking passwords during the management of sessions, and the security setting function (**SF.MGMT**) for checking the complexity of passwords. For both functions, the strength of security is SOF-basic.

6.3. Assurance Measures

Table 7 describes the correspondence between the security assurance requirements and the security assurance measures applied in this ST. The documents and products listed in the table are provided as the security assurance measures to be applied in this ST.

Table 7: Correspondence between security assurance requirements and security assurance measures

Security assurance requirement		Security assurance measure
ACM_CAP.2	Configuration items	HiCommand Suite Common Component Configuration Management Document
ADO_DEL.1	Delivery procedures	HiCommand Suite Common Component Delivery Document
ADO_IGS.1	Installation, generation, and start-up procedures	HiCommand Suite Common Component Security Guide
ADV_FSP.1	Informal functional specification	HiCommand Suite Common Component Functional Specifications
ADV_HLD.1	Descriptive high-level design	HiCommand Suite Common Component Structure Design
ADV_RCR.1	Informal correspondence demonstration	HiCommand Suite Common Component Correspondence Analysis
AGD_ADM.1	Administrator guidance	HiCommand Suite Common Component Security Guide
AGD_USR.1	User guidance	HiCommand Suite Common Component Security Guide
ALC_FLR.1	Basic flaw remediation	HiCommand Suite Common Component Security Flaw Remediation Specifications
ATE_COV.1	Evidence of coverage	HiCommand Suite Common Component Test Design
ATE_FUN.1	Functional testing	HiCommand Suite Common Component Test Report
ATE_IND.2	Independent testing - sample	HiCommand Suite Common Component 05-51
AVA_SOF.1	Strength of TOE security function evaluation	HiCommand Suite Common Component Security Function Strength Analysis
AVA_VLA.1	Developer vulnerability analysis	HiCommand Suite Common Component Vulnerability Analysis

7. Protection Profile (PP) Claims

There are no Protection Profile claims in this Security Target.

8. Rationale

This section describes the rationale for the security objectives, security requirements, and TOE summary specification.

8.1. Security Objectives Rationale

The security objectives counter the threats specified in the TOE security environment, uphold assumptions, and enforce organizational security policies. Table 8 describes the correspondence among security objectives, the threats to be countered, assumptions to be upheld, and organizational security policies to be enforced.

Table 8: Correspondence among security objectives, assumptions, threats, and organizational security policies

TOE security environment Security objective	A.PHYSICAL	A.NETWORKS	A.ADMINISTROTORS	A.SECURE_CHANNEL	A.TOKEN	A.PASSWORD	T.ILLEGAL_ACCESS	T.UNAUTHORISED_ACCESS	P.BANNER
O.I&A							X		
O.MGMT								X	
O.BANNER									X
O.PASSWORD							X		
OE.SECURE_CHANNEL				X					
OE.BANNER									X
OM.PHYSICAL	X								
OM.FIREWALL		X							
OM.ADMINISTROTORS			X						
OM.TOE_ACCOUNT							X		
OM.TOKEN					X				
OM.PASSWORD						X			

As shown in Table 8, each security objective corresponds to at least one assumption, threat, or organizational security policy.

The following describes how security objectives counter threats, uphold assumptions, and enforce organizational security policies.

(1) Security threats

T.ILLEGAL_ACCESS (illegal connection)

O.I&A ensures that the TOE identifies and authenticates a user who accesses the TOE and storage management software, and checks whether the user is authorized. **O.PASSWORD** ensures that the TOE limits the registration patterns of passwords so that the passwords that are set cannot be guessed easily. **OM.TOE_ACCOUNT** ensures that a user does not reveal to others the password the user has created. A password that can be set is difficult to guess and is changed at an adequate frequency, making it difficult for illegal users to learn other users' passwords.

T.ILLEGAL_ACCESS is therefore countered by **O.I&A**, **O.PASSWORD** and **OM.TOE_ACCOUNT**.

T.UNAUTHORISED_ACCESS (unauthorized access)

O.MGMT ensures that the TOE controls access to permissions and banner information by users based on the permissions granted to the users of the storage management software and the TOE.

T.UNAUTHORISED_ACCESS is therefore countered by **O.MGMT**.

(2) Assumptions

A.PHYSICAL (management of hardware)

OM.PHYSICAL ensures that the management server running the TOE and the storage management software, peripheral devices, the storage devices, the internal network, and the firewall installed at the boundary of the internal network are installed in the physically isolated business server area. Entry and exit to and from the business server area are controlled so that only authorized administrators can enter the area.

A.PHYSICAL is therefore upheld by **OM.PHYSICAL**.

A.NETWORKS (networks)

OM.FIREWALL ensures that a firewall is installed between the internal network in the business server area containing the management network connected to the management server and the external network in order to stop unnecessary communication and remote operations from the external network to the TOE in the internal network. Each network is logically separated and traffic is monitored to detect illegal traffic.

A.NETWORKS is therefore upheld by **OM.FIREWALL**.

A.ADMINISTRATORS (administrators)

OM.ADMINISTRATORS ensures that those with highest level of responsibility in an organization select appropriate personnel for the system builder, account administrators, storage administrators, and administrators of other servers, including business servers. Therefore, the system builder is a trusted person. Also, account administrators, storage administrators, and the administrators of other servers, including business servers, do not perform malicious acts regarding one another's work. Work includes the management of the accounts and permissions of storage management software users, the management of storages, and the management of other servers.

A.ADMINISTRATORS is therefore upheld by **OM.ADMINISTRATORS**.

A.SECURE_CHANNEL (communication secrecy)

OE.SECURE_CHANNEL ensures that the network between the management server and management clients uses communication paths protected by encryption or other methods to ensure the secrecy and completeness of communication.

A.SECURE_CHANNEL is therefore upheld by **OE.SECURE_CHANNEL**.

A.TOKEN (available tokens)

OM.TOKEN ensures that the system builder does not create an environment with products that use the following tokens and the TOE:

- Tokens that are generated by an entity other than the TOE
- Tokens from which user IDs and user passwords can be guessed

A.TOKEN is therefore upheld by **OM.TOKEN**.

A.PASSWORD (complex passwords)

OM.PASSWORD ensures that administrators specify settings that require complex passwords and limit the number of repeated authentication attempts, thereby preventing logon by illegal users who have guessed passwords.

A.PASSWORD is therefore upheld by **OM.PASSWORD**.

(3) Organizational security policies

P.BANNER (warning banner)

O.BANNER ensures that the TOE provides storage management software with advisory warning messages regarding illegal use of storage management software. **OE.BANNER** ensures that storage management software has functionality for displaying advisory messages (provided by the TOE) regarding illegal use of storage management software.

P.BANNER is therefore enforced by **O.BANNER** and **OE.BANNER**.

8.2. Security Requirements Rationale

8.2.1. TOE Security Functional Requirements Rationale

Table 9 describes the relation between the security functional requirements for the TOE and for the IT environment selected in this ST and the security objectives for the TOE.

Table 9: Relation between security functional requirements and TOE security objectives

TOE security objective / TOE security functional requirement	O.I&A	O.MGMT	O.BANNER	O.PASSWORD	OE.SECURE_CHANNEL	OE.BANNER
FDP_ACC.1		X	X			
FDP_ACF.1		X	X			
FMT_MSA.1		X				
FMT_MSA.3		X				
FMT_MTD.1	X	X				
FMT_SMF.1		X				
FMT_SMR.1		X				
FIA_UAU.1	X					
FIA_UID.1	X					
FIA_SOS.1				X		
FIA_ATD.1	X					
FIA_USB.1	X					
FIA_AFL.1	X					
FIA_TAB.1			X			
FPT_RVM.1	X	X	X			
FPT_SEP.1	X	X				
FPT_ITC.1					X	
FIA_TAB.1E						X

As shown in Table 9, each security functional requirement for the TOE corresponds to at least one TOE security objective, and each security functional requirement for the IT environment corresponds to at least one security objective for the IT environment.

The following describes how each security objective for the TOE can be achieved by the security functional requirements for the TOE.

O.I&A

When a user accesses the TOE and the storage management software, the TOE uses **FIA_UID.1** to identify whether the user is authorized and uses **FIA_UAU.1** to authenticate that the user is really an authorized user. At this time, the TOE uses **FIA_AFL.1** to lock the account of a user whose failed authentication attempts have reached the preset number of times. The TOE uses **FIA_ATD.1** to maintain and manage the user IDs and roles of users who are successfully identified and authenticated as session data and uses **FIA_USB.1** to associate the user IDs and roles with the processes that act on behalf of the identified and authenticated users.

The TOE also uses **FMT_MTD.1** so that only account administrators and the system builder can manage the user ID, password, and lock status of each user.

The TOE uses **FPT_RVM.1** and **FPT_SEP.1** to prevent bypassing of and interference and tampering with the security functions.

O.I&A is therefore achieved with **FIA_UAU.1**, **FIA_UID.1**, **FIA_ATD.1**, **FIA_AFL.1**, **FIA_USB.1**, **FMT_MTD.1**, **FPT_RVM.1**, and **FPT_SEP.1**.

O.MGMT

The TOE uses **FMT_MSA.1** to allow only account administrators and the system builder to manage the user IDs and roles that are the security attributes of users. The TOE also uses **FMT_MSA.3** to provide restrictive initial values at a time when roles have not been set. The TOE uses **FMT_MTD.1** to allow only account administrators and the system builder to manage security parameters.

When the TOE uses **FDP_ACC.1** and **FDP_ACF.1** to acquire the role and permission of a successfully authenticated user from the ACL table, the TOE controls access to the ACL table based on the user ID of the user. If the user attempts to manipulate the ACL table and the banner information file, the TOE controls access to the ACL table and the banner information file based on the user ID and role of the user.

The TOE uses **FMT_SMR.1** to maintain the storage administrator, account administrator, and system builder roles.

The TOE uses **FMT_SMF.1** so that it can execute the security management functions indicated by management items.

The TOE also uses **FPT_RVM.1** and **FPT_SEP.1** to prevent bypassing of and interference and tampering with the security functions.

O.MGMT is therefore achieved by **FDP_ACC.1**, **FDP_ACF.1**, **FMT_MSA.1**, **FMT_MSA.3**,

FMT_MTD.1, **FMT_SMF.1**, **FMT_SMR.1**, **FPT_RVM.1**, and **FPT_SEP.1**.

O.BANNER

The TOE uses **FIA_TAB.1** to acquire an advisory warning message regarding illegal use of the storage management software and passes this message to the storage management software. At this time, the TOE uses **FDP_ACC.1** and **FDP_ACF.1** to control access to the banner information file so that viewing of the banner information file containing the warning message is always allowed.

The TOE also uses **FPT_RVM.1** to prevent bypassing of the access control described above.

O.BANNER is therefore achieved by **FIA_TAB.1**, **FDP_ACC.1**, **FDP_ACF.1**, and **FPT_RVM.1**.

O.PASSWORD

The TOE uses **FIA_SOS.1** to maintain the quality criteria for secrets (passwords).

O.PASSWORD is therefore achieved by **FIA_SOS.1**.

The following describes how the security objectives for the IT environment are achieved by the security functional requirements for the IT environment.

OE.SECURE_CHANNEL

The TOE uses **FPT_ITC.1** to request that the network between the management server and management clients provide protected communication paths by encryption or other means in order to protect communication data such as user IDs, passwords, and tokens from changes or exposure.

OE.SECURE_CHANNEL is therefore achieved by **FPT_ITC.1**.

OE.BANNER

The TOE uses **FIA_TAB.1E** to request that the storage management software display an advisory warning message regarding illegal use of the storage management software.

OE.BANNER is therefore achieved by **FIA_TAB.1E**.

8.2.2. Rationale for the minimum strength of function level

The threats assumed by this TOE are low-level agents without advanced expertise that use the interface of the clients that are operated by administrators. Therefore, SOF-basic is appropriate as the minimum strength of the function level. This level matches the minimum strength of the function level that is required by the ST for the TOE.

8.2.3. Dependencies of security functional requirements

Table 10 describes the dependencies of the components of the security functional requirements.

Table 10: Dependencies of the components of the security functional requirements

Functional requirement component selected in this ST	Dependent component specified in CC Part 2	Dependent component selected in this ST	Whether achieved
FDP_ACC.1	**FDP_ACF.1**	**FDP_ACF.1**	X
FDP_ACF.1	**FDP_ACC.1**	**FDP_ACC.1**	X
	FMT_MSA.3	**FMT_MSA.3**	X
FMT_MSA.1	**FDP_ACC.1**	**FDP_ACC.1**	X
	FMT_SMF.1	**FMT_SMF.1**	X
	FMT_SMR.1	**FMT_SMR.1**	X
FMT_MSA.3	**FMT_MSA.1**	**FMT_MSA.1**	X
	FMT_SMR.1	**FMT_SMR.1**	X
FMT_MTD.1	**FMT_SMF.1**	**FMT_SMF.1**	X
	FMT_SMR.1	**FMT_SMR.1**	X
FMT_SMF.1	None	**-**	**-**
FMT_SMR.1	**FIA_UID.1**	**FIA_UID.1**	X
FIA_UAU.1	**FIA_UID.1**	**FIA_UID.1**	X
FIA_UID.1	None	**-**	**-**
FIA_SOS.1	None	**-**	**-**
FIA_ATD.1	None	**-**	**-**
FIA_USB.1	**FIA_ATD.1**	**FIA_ATD.1**	X
FIA_AFL.1	**FIA_UAU.1**	**FIA_UAU.1**	X
FIA_TAB.1	None	**-**	**-**
FPT_RVM.1	None	**-**	**-**
FPT_SEP.1	None	**-**	**-**
FPT_ITC.1	None	**-**	None
FIA_TAB.1E	None	**-**	None

Each security functional requirement therefore satisfies all necessary dependencies.

8.2.4. Dependencies of security assurance requirements

ALC_FLR.1 does not depend on an assurance component.

8.2.5. Internal consistency and mutual support of security functional requirements

All the dependent TOE security functional requirements are selected, and no insufficiency exists in achieving the characteristics of the functional requirements.

Preventing the bypassing of security functions :

FPT_RVM.1 is selected for access control, identification and authentication, and the security information management function to prevent the security functions from being bypassed. Since no overlapping functional requirements are selected, no conflict or contention occurs.

Preventing interference with the security functions :

As a defense against interference, FPT_SEP.1 is selected to protect the TSF and the TSF data.

Disabling :

The security functions are enabled as soon as the TOE is installed. The TOE built-in account (system builder account) assumes that it will use the enabled security functions. Because the security functions are never disabled in this TOE, FMT_MOF.1 does not need to be selected.

8.2.6. Rationale for audit

This TOE claims FIA_SOS.1 as one of the functional requirements. The TOE provides mechanisms for verifying that a password for identifying and authenticating a user satisfies the complex password condition and the minimum number of characters required in a password. OM.PASSWORD ensures the setting of security parameters that allow only hard-to-guess passwords to be used. As an assumption for using the TOE, trusted users are required to set hard-to-guess passwords in the TOE.

This TOE also claims FIA_AFL.1 as a functional requirement. The TOE provides functionality that locks an account if repeated authentication attempts fail. OM.PASSWORD ensures that security parameters that limit the number of repeated logon attempts are set.

Accordingly, this ST does not specify security objectives that use an audit to detect repeated logon attempts by users that are not registered in the TOE. Since security functional requirement FAU_GEN.1 is not selected, the rationale for audit events is not applicable.

8.2.7. Rationale for management requirements

Table 3 describes the relation between the management requirements specified in CC Part 2 and the management items to be managed by the TSF regarding the TOE security functional requirements selected in this ST.

Table 11 describes the relation between the TSF management items in Table 3 and the TOE security functions described in Section 6.1.

Table 11: Relation between TSF management items and TOE security functions

Functional requirement	Management item	TOE security function
FDP_ACC.1	None	—
FDP_ACF.1	a) Management of user IDs and associated permissions	a) **SF.MGMT**
FMT_MSA.1	a) None (no groups of roles that may affect security attributes that may affect roles exist)	a) -
FMT_MSA.3	a) None (no groups of roles exist) b) None (no management of default value settings exists)	a) - b) **SF.MGMT**

FMT_MTD.1	a) None (no groups of roles that may affect TSF data that may affect roles exist)	a) -
FMT_SMR.1	a) None (no groups of users that consist of parts of roles exist)	a) -
FIA_UAU.1	a) Creation and change of passwords b) Change of passwords by users c) None (lists are not changed)	a) **SF.MGMT** b) **SF.MGMT** c) -
FIA_UID.1	a) Creation and deletion of user IDs for accounts b) None (lists are not changed)	a) **SF.MGMT** b) -
FIA_SOS.1	a) Specification of the required number of characters and types of characters in passwords when passwords are set	a) **SF.MGMT**
FIA_ATD.1	a) None (no additional security attributes are defined)	a) -
FIA_USB.1	a) None (no security attributes are given by default) b) None (since no security attributes are given by default)	a) - b) -
FIA_AFL.1	a) Setting and changing of threshold values by administrators b) None (the only action to be performed is locking accounts)	a) **SF.MGMT** b) -
FTA_TAB.1	a) Setting of banner contents by administrators	a) **SF.MGMT**
FPT_RVM.1	None	-
FPT_SEP.1	None	-

8.2.8. Rationale for security assurance requirements

The evaluation assurance level of this TOE is EAL2 with ALC_FLR.1.

The users assumed by this TOE are storage administrators. Their number is limited, and each is registered. Therefore, intent to attack is suppressed. EAL2 is the appropriate choice because it includes evaluation from the point of view of structural design, secure delivery procedures, and vulnerability assessment for the TOE with the described characteristics.

Recently, finding means to handle problems related to security vulnerability has become important. This product plays an important part in managing storages, and is required to trace security flaws and act quickly when vulnerability problems arise. Because assurance in the face of security flaws is important in providing safety for users, ALC_FLR.1 is selected.

8.3. TOE Summary Specification Rationale

8.3.1. TOE security functions Rationale

Table 12 describes the relation between the TOE security functions and the TOE security functional requirements.

Table 12 Relation between TOE security functions and TOE security functional requirements

TOE security functional requirement TOE security function	FDP_ACC.1	FDP_ACF.1	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FIA_UAU.1	FIA_UID.1	FIA_SOS.1	FIA_ATD.1	FIA_USB.1	FIA_AFL.1	FIA_TAB.1	FPT_RVM.1	FPT_SEP.1
SF.I&A	X	X						X	X		X	X	X		X	X
SF.MGMT	X	X	X	X	X	X	X			X					X	X
SF.BANNER	X	X												X	X	

As shown in Table 12, each TOE security function has at least one TOE security functional requirement.

The following proves that each TOE security functional requirement is achieved by the TOE security functions.

FDP_ACC.1:

FDP_ACF.1:

When the process (subject) that identifies and authenticates a user on behalf of the user reads the ACL table (object) to obtain the role and permission granted to the user, the TOE uses **SF.I&A** to control access to the object based on the user ID associated with the subject and the user ID in the object.

When the process (subject) that acts on behalf of a user reads, changes, creates, or deletes the ACL table or the banner information file (object), the TOE uses **SF.MGMT** to control access to the object based on the user ID and the role associated with the subject and the user ID in the object.

When the process (subject) that acts on behalf of a user reads the banner information file (object) to acquire a warning message, the TOE uses **SF.BANNER** to control access to permit only read accesses.

FDP_ACC.1 and **FDP_ACF.1** are therefore achieved for **SF.I&A**, **SF.MGMT**, and **SF.BANNER**.

FMT_MSA.1:

The TOE uses **SF.MGMT** to allow only account administrators and the system builder to change and delete the user ID and role that are associated with the object (ACL table) and that are security attributes. However, account administrators are not allowed to change their own role and the role for the system builder account.

FMT_MSA.1 is therefore achieved for **SF.MGMT**.

FMT_MSA.3:

The TOE uses **SF.MGMT** to give restrictive initial values to the roles corresponding to user IDs that are security attributes when roles are not set.

FMT_MSA.3 is therefore achieved for **SF.MGMT**.

FMT_MTD.1:

The TOE uses **SF.MGMT** to provide a function that manages the user ID (account), password, lock status, and security parameters of each user. The TOE allows only account administrators and the system builder to register and delete user IDs, to register, change, and delete passwords (which deletes entire accounts), to query and change the lock status, and to query, change, and clear security parameters.

Note that the TOE allows storage administrators to change their own passwords.

The TOE cannot register or delete the user ID for the system builder account.

FMT_MTD.1 is therefore achieved for **SF.MGMT**.

FMT_SMF.1:

As described in Subsection 8.2.6, among the requirements specified in CC Part 2 that need to be managed for the functional requirements selected in this ST, **SF.MGMT** manages the items that are to be managed by the TOE.

FMT_SMF.1 is therefore achieved for **SF.MGMT**.

FMT_SMR.1:

The TOE uses **SF.MGMT** to maintain the storage administrator, account administrator, and system builder roles, to associate each role with a user, and to manage them in the ACL table.

FMT_SMR.1 is therefore achieved for **SF.MGMT**.

FIA_UAU.1, FIA_UID.1:

Until **SF.I&A** is able to identify and authenticate users, no action is executed except for sending a warning message provided by the warning banner function (**SF.BANNER**). .

FIA_UAU.1 and **FIA_UID.1** are therefore achieved for **SF.I&A**.

FIA_SOS.1:

When a new account is created or when a password is registered or changed, the TOE uses **SF.MGMT** to provide mechanisms for verifying that the password satisfies the following quality criteria:

・ The password satisfies the minimum number of characters required in a password, which is determined in a security parameter.

- The types of characters allowed in a password are alphabetic and numeric characters and symbols, and the complex password condition determined in a security parameter is satisfied.

FIA_SOS.1 is therefore achieved for **SF.MGMT**.

FIA_ATD.1, **FIA_USB.1:**

The TOE uses **SF.I&A** to maintain and manage the user ID and role of a user who has been successfully identified and authenticated, and to associate the user ID and role with the process that acts on behalf of the user.

FIA_ATD.1 is therefore achieved for **SF.I&A**.

FIA_AFL.1:

When the TOE authenticates a user who logs on to the storage management software, the TOE uses **SF.I&A** to lock the account of a user whose failed authentication attempts has reached the predefined number of times.

FIA_AFL.1 is therefore achieved for **SF.I&A**.

FIA_TAB.1:

The TOE uses **SF.BANNER** to send an advisory warning message regarding illegal use of the storage management software to the storage management software. The storage management software displays that warning message in the logon window used to identify and authenticate users.

FIA_TAB.1 is therefore achieved for **SF.BANNER**.

FPT_RVM.1:

SF.I&A ensures that it is always executed when it receives a request for identifying and authenticating a user from the storage management software.

SF.I&A ensures that access control is always performed when the process (subject) acting on behalf of the user described above accesses the ACL table (object).

SF.MGMT ensures that access control is always performed when the process (subject) acting on behalf of a user accesses the ACL table (object) and the banner information file (object).

SF.BANNER ensures that access control is always performed when the process (subject) acting on behalf of a user accesses the banner information file (object).

FPT_RVM.1 is therefore achieved for **SF.I&A**, **SF.MGMT**, and **SF.BANNER**.

FPT_SEP.1:

Since the user ID and role associated with each process that acts on behalf of a successfully logged-on user are separately located in a security domain that allows only access from authorized processes, **SF.I&A** ensures that untrusted processes cannot change user IDs and roles. Untrusted processes are all processes other than the process that acts on behalf of the user who is successfully logged on.

Since the information in the ACL is separately located in a security domain that allows only access from authorized processes, **SF.MGMT** ensures that untrusted processes cannot change the information in the ACL. Untrusted processes are all processes other than the process that acts on behalf of the user who is successfully identified and authenticated.

Since banner information is separately located in a security domain that allows access only from the process that is authorized to use the banner information file editing function and the system builder that has successfully logged on to the management server, **SF.MGMT** ensures that untrusted processes cannot change the banner information. Untrusted processes are all processes other than the process that acts on behalf of the user who is successfully identified and authenticated.

FPT_SEP.1 is therefore achieved for **SF.I&A** and **SF.MGMT**.

8.3.2. Strength of Functions Rationale

For this TOE, the security functions that are based on a probabilistic or permutational mechanism are the token generation mechanism and password matching mechanism of **SF.I&A**, and the password complexity checking mechanism of **SF.MGMT**. For the strength of these security functions, Subsection 6.2 specifies SOF-basic. As the minimum strength of function level in this TOE, Subsection 5.1.2 specifies SOF-basic. Accordingly, the strength is consistent.

8.3.3. Assurance Measures Rationale

This subsection explains that security assurance measures are necessary and sufficient for the security assurance requirements. Table 7 describes the relation between the security assurance requirements and the security assurance measures.

As shown in Table 7, the listed security assurance measures are required for particular security assurance requirements. The contents of the security measures (documents) cover all the evidence requested by the security assurance requirements specified in this ST.

Therefore, the security assurance measures applied in this ST can satisfy the security assurance requirements.

8.4. PP Claims Rationale

No PP was referenced.