UbiNtisLab

Pass-Ni SSO v4.0

Security Target

V1.0 R6

UbiNtisLab Co., Ltd

Revision History

Ver	Data	Detail	Author
R1	2017.12.20	Initial release	Research Institute
R2	2018.08.16	TOE Updated (v4.0.002)	Research Institute
R3	2018.09.14	TOE Updated (v4.0.003)	Research Institute
R4	2018.11.02	TOE Updated (v4.0.004)	Research Institute
R5	2019.01.03	TOE Updated (v4.0.005)	Research Institute
R6	2019.02.18	ST Updated	Research Institute

Table of Contents

1. ST Introduction

This chapter introduces the Security Target (ST) of Pass-Ni SSO v4.0 of UbiNtisLab Co., Ltd.

1.1 ST Reference

Item	Specification
Title	Pass-Ni SSO v4.0 Security Target
Version	V1.0 R6
Author	UbiNtisLab Co., Ltd Research Institute
Publication Date	2019-02-18
Configuration Management No.	PassNi-SSO-v4.0-ST-V1.0.R6
Common Criteria	Common Criteria for Information Technology Security Evaluation V3.1R5
Protection Profile	National Protection Profile for Single Sign-On V1.0
Evaluation Assurance Level	EAL1+(ATE_FUN.1)

[Table 1-1] ST Reference

1.2 TOE Reference

Item	Specification
TOE	Pass-Ni SSO v4.0
TOE Version	v4.0.005
TOE Components	Pass-Ni SSO Server v4.0.005 Pass-Ni SSO Agent for JAVA v4.0.005 Pass-Ni SSO Agent for ASP.NET v4.0.005 Pass-Ni SSO Agent for PHP v4.0.005
Developer	UbiNtisLab Co., Ltd

[Table 1-2] TOE Reference

The TOE uses the following validated cryptographic module.

Item	Specification
Cryptographic module name	Pass-Ni Crypto V1.1
Validation number	CM-123-2021.12
Validation date	2016-12-05
Expiration date	2021-12-05
Developer	UbiNtisLab Co., Ltd

[Table 1-3] Cryptographic Module Reference

1.3 TOE Overview

This section prescribes the usage of the TOE and major security features. It also identifies types of the TOE and identifies software, hardware and firmware required by the TOE but not the TOE.

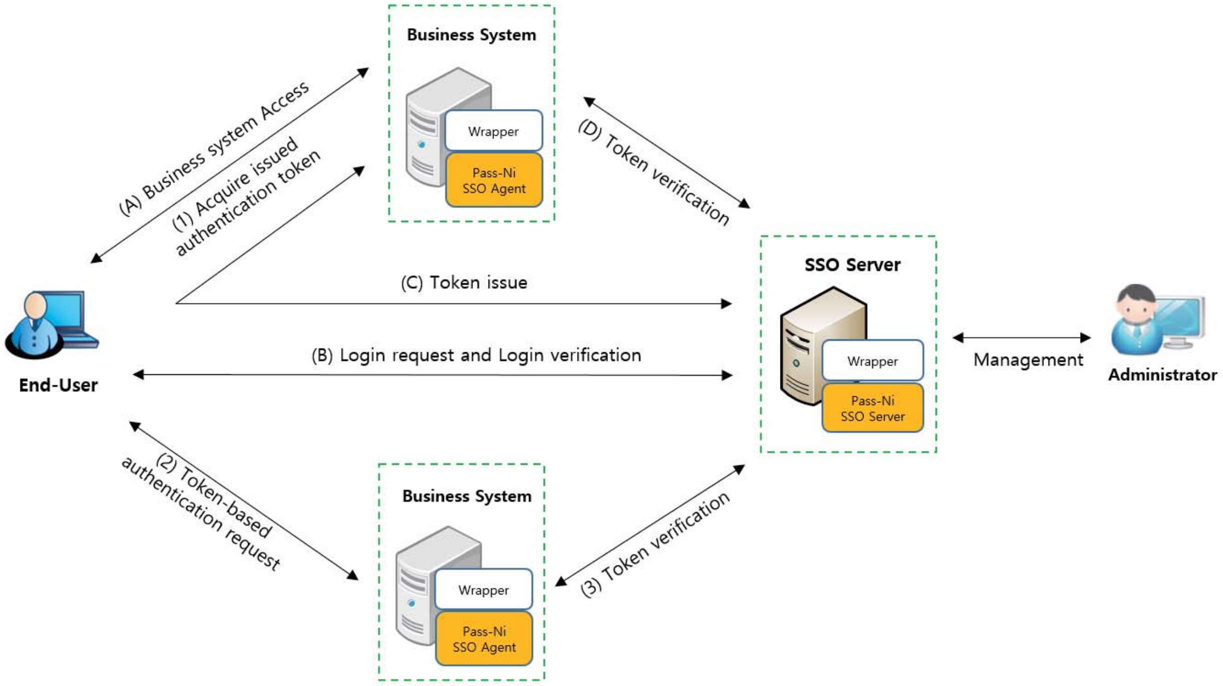
1.3.1 TOE Usages and Major Security Features

The TOE is an ‘Single Sign On (SSO)' that is used for the purpose of providing a user with services from various application servers (business systems) without additional login by single login. The TOE provides users with access to information of various business systems through single authentication.

The major features of the TOE are to issue, store, verify, and revoke the authentication token. It issues an authentication token when the user, who requests the login, is regarded as valid by identifying and authenticating. Then, when the user accesses the other business system, the access of the user is controlled through validation of the authentication token. In the initial authentication phase, the TOE performs ID / PW authentication for doing identification and authentication functionality.

TOE security functions include **the security audit function** that manages and records major cases as audit data, **the identification and authentication function** for users, **TSF protection functions** such as TSF data protection function and TSF self test. It also includes **the cryptographic support function** for performing cryptographic key management and cryptographic operations, **the security management function** for security policy and environment setting, and **the TOE access function** for controlling connection sessions of the authorized administrator.

The end-user identification and authentication process is divided into the initial authentication phase using the ID/password and the token-based authentication phase that accesses the business system using the token issued during the initial authentication procedure.



[Figure 1-1] End-user identification and authentication procedure

The procedure of the initial authentication step is as follows.
When the end-user accesses the business system, the Pass-Ni SSO Agent verifies the user's authentication status, and if not, moves to the Pass-Ni SSO Server's login page. End-user requests login verification through ID/PW and The Pass-Ni SSO Server performs login verification using general user information stored in DBMS. The Pass-Ni SSO Server issues an authentication token to the Pass-Ni SSO Agent via the end-user's browser if the login verification result is valid. The Pass-Ni SSO Agent verifies the authentication token from the Pass-Ni SSO Server.

The token-based authentication phase is performed only when the token has been normally issued in the initial authentication phase. The end-user acquires the authentication token through the business system that has received the authentication token in the initial authentication step and requests token-based authentication to the SSO target business system. The Pass-Ni SSO Agent verifies the authentication token from the Pass-Ni SSO Server.

Authentication phase	Operation procedure
Initial authentication	(A) Business system Access – (B) Login request and Login verification – (C) Token issue – (D) Token verification
Token-based authentication	(1) Acquire issued authentication token – (2) Token-based authentication request – (3) Token verification

[Table 1-4] Operation procedure by authentication phase

- Authentication token issuer: Pass-Ni SSO Server
- Authentication token storage location: Pass-Ni SSO Server, Pass-Ni SSO Agent
- Authentication token validator: Pass-Ni SSO Server

1.3.2 TOE Types

The TOE is offored in the form of software which is 'Single Sign On (SSO)' that allows access to various application servers (business systems) through a single user login.

The TOE component comprised of the SSO server that performs functions such as processing user login, issuing authentication token and managing policy, and the SSO agent that is installed in each business system and verifies the validity of the authentication token through interworking with the SSO server. The SSO agent is provided as 'API' consisting of library files.

1.3.3 (Non-TOE) Hardware/Software/Firmware, Out of TOE required evaluation target

The TOE, in the form of software, is installed in a server or a PC and operated on an operating system (OS) such as Windows or Linux. The hardware / software, which is out of the TOE evaluation target, required for TOE server operation is identified as follows.

Type		Requirements for TOE Server
H/W	CPU	Intel® Xeon^{TM} E5 2.0Ghz higher
	HDD	Space required for installation of TOE 50GB or higher
	Memory	8GB higher
	NIC	Ethernet 100/1000 Mbps * 1port higher
OS		Windows Server 2008 Standard x86 Windows Server 2012 Standard x64 CentOS 6.9 (kernel 2.6.32) x64 Ubuntu Server 16.04 (kernel 4.4.0) x64
S/W		jdk-8u152 Apache Tomcat 8.5.23 CUBRID 9.3.9

[Table 1-5] Hardware/Software Requirements for TOE Server

The hardware / software, which is out of the TOE evaluation target, required for the TOE Agent operation is identified as follows.

Agent type	Type		Requirements for TOE Agent
Common	H/W	CPU	Intel® Core^{TM} i5 2.6Ghz higher
		HDD	Space required for installation of TOE 10GB or higher
		Memory	8GB higher
		NIC	Ethernet 100/1000 Mbps * 1port higher
Agent for JAVA	OS		Windows Server 2008 Standard x86 Windows Server 2012 Standard x64 CentOS 6.9 (kernel 2.6.32) x64 Ubuntu Server 16.04 (kernel 4.4.0) x64
	S/W		jdk-8u152
Agent for ASP.NET	OS		Windows Server 2012 Standard x64
	S/W		.NET Framework 4.7.1
Agent for PHP	OS		CentOS 6.9 (kernel 2.6.32) x64
	S/W		PHP 5.3.3

[Table 1-6] Hardware/Software Requirements for TOE Agent

The hardware / software requirements of the PC used by the authorized administrator to manage the TOE are as follows.

Type		Requirements for administrator's PC
H/W	CPU	Intel® Core^{TM}2 Duo 1.6Ghz higher
	HDD	500GB higher
	Memory	4GB higher
	NIC	Ethernet 100/1000 Mbps * 1port higher
OS		Windows 10 Pro x64
Browser		Internet Explorer 11 Google Chrome 65

[Table 1-7] Hardware/Software Requirements for administrator's PC

The major roles of hardware / software other than the evaluation target required for the TOE operation are as follows.

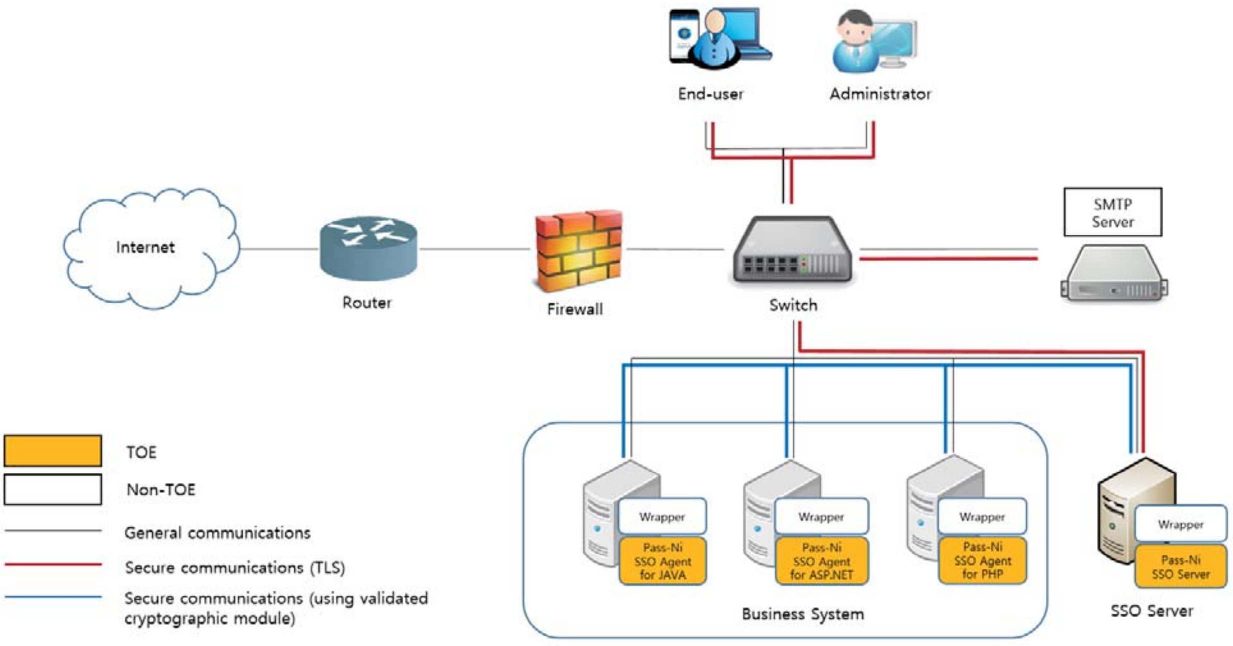
- H/W and OS: Provides operational environment that ensures the reliability and availability of the TOE.
- CUBRID 9.3.9: Records and stores security policies required for the operation of the TOE and security audit data generated during the operation of the TOE
- Apache Tomcat 8.5.23: The web application server to provide security features and management
- jdk-8u152: The runtime platform for running Java applications
- .NET Framework 4.7.1: The runtime platform for running .NET applications
- PHP 5.3.3: The runtime platform for running PHP applications
- Internet Explorer 11: The web browser for accessing TOE security management interface
- Google Chrome 65: The web browser to access TOE security management interface

1.4 TOE Description

This section prescribes the TOE operational environment, the physical scope and the logical scope.

1.4.1 TOE Operational Environment

The operational environment of the TOE is shown in the following figure.


End-user
Administrator
Internet
Router
Firewall
Switch
SMTP Server
TOE
Non-TOE
General communications
Secure communications (TLS)
Secure communications (using validated cryptographic module)
Wrapper
Pass-Ni SSO Agent for JAVA
Wrapper
Pass-Ni SSO Agent for ASP.NET
Wrapper
Pass-Ni SSO Agent for PHP
Business System
Wrapper
Pass-Ni SSO Server
SSO Server

[Figure 1-2] TOE Operational environment

The TOE comprised of the Pass-Ni SSO server that performs functions such as processing user login, issuing authentication token and managing policy, and the Pass-Ni SSO agent that is installed in each business system and verifies the validity of the authentication token.

The major roles of the operational services other than the TOE evaluation target required for the TOE operation are as follows.

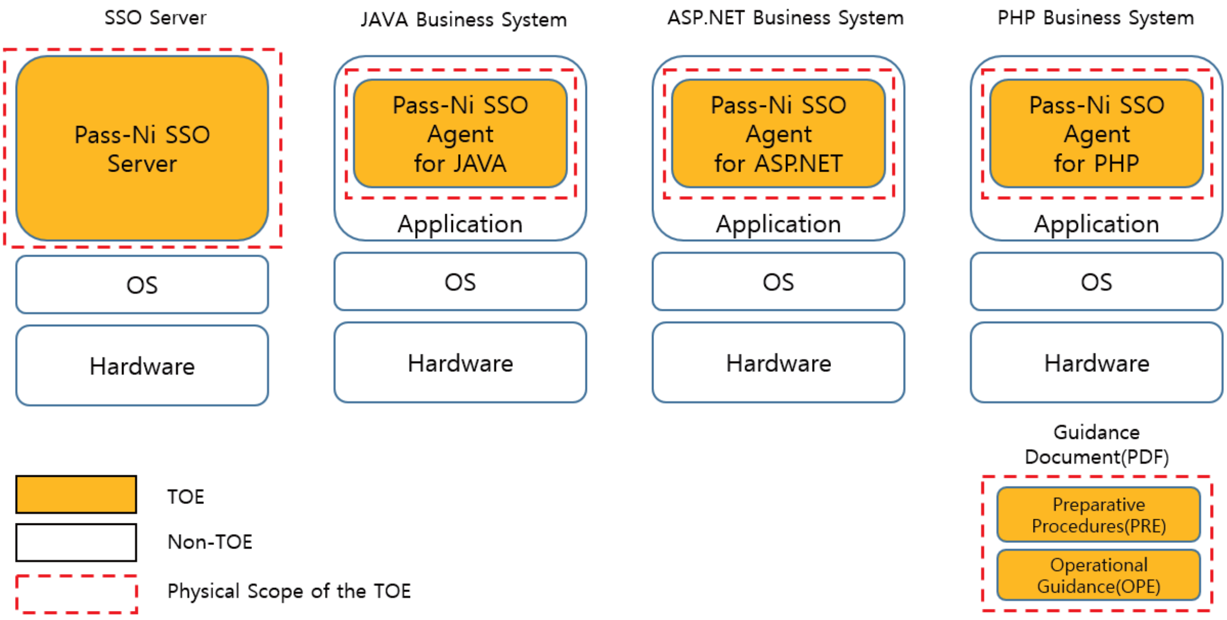
- SMTP server: The mail server that sends an administrator notification mail, such as handling authentication failures and saturation of audit storage

1.4.2 The physical scope of the TOE

The physical scope of the TOE consists of **Pass-Ni SSO Server** which performs functions such as user login processing, authentication token issuance and policy setting, and **Pass-Ni SSO Agent** which is installed in each business system and validates the authentication token through interworking with SSO server. It also includes **preparative procedures** that describe the procedures for secure acceptance and installing the TOE, and **operational guidance** that specify how to use the TOE safely.

The hardware and operating system where the TOE is installed, an administrator PC connect as privileged mode for the TOE security management, the DBMS storing the security policy and audit data, and the Wrappers which may be used to support various types of compatibility with business systems are excluded from the TOE physical scope.

The physical scope of the TOE is shown in the following figure.



[Figure 1-3] Physical Scope of the TOE

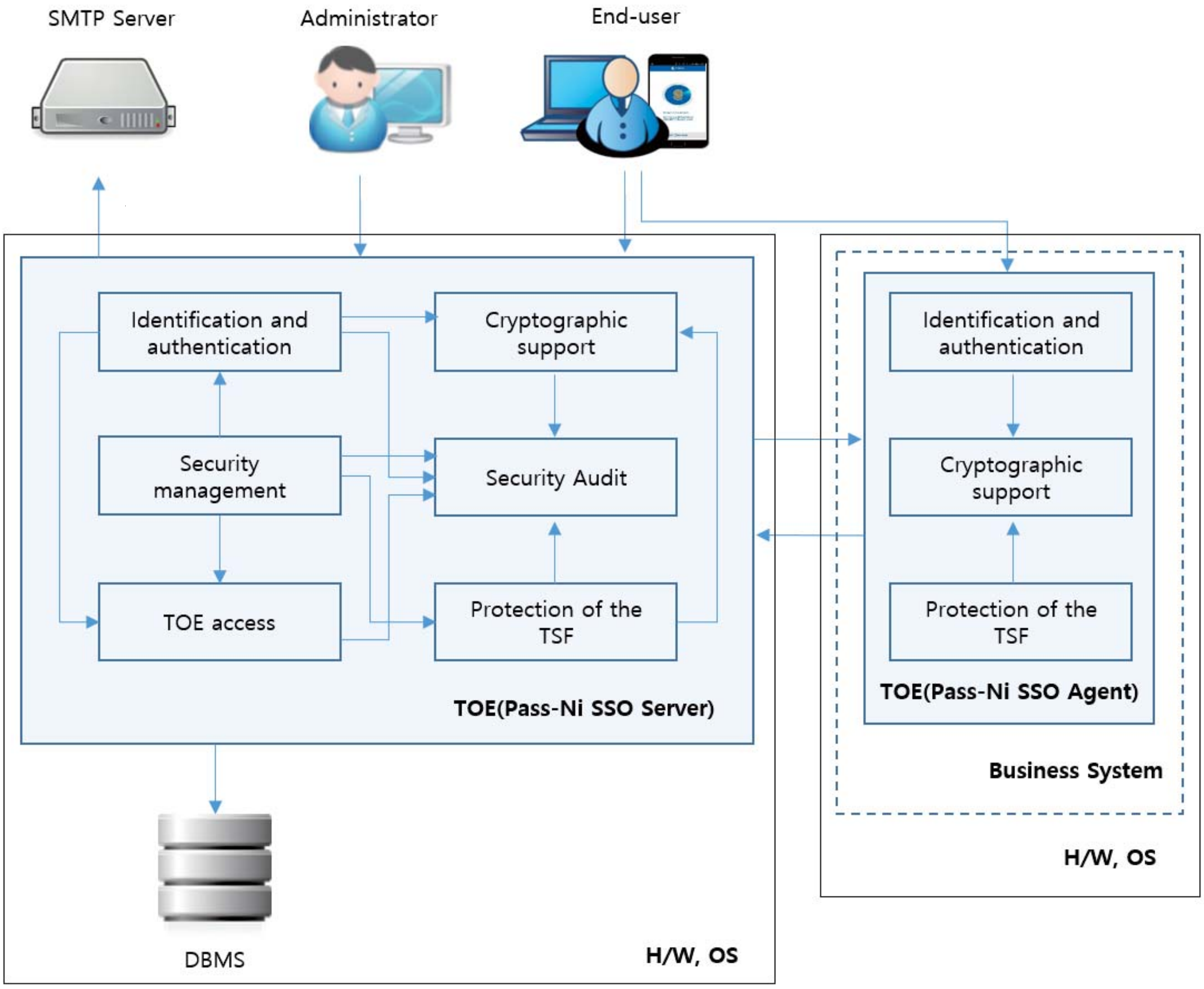
The TOE distributed by the purchaser is distributed as files on the package CD and comprised of the following components.

Item	Type	Element
S/W	Installation Package	Pass-Ni SSO Server v4.0.005 (PassNi-SSO-Server-v4.0.005.zip)
		Pass-Ni SSO Agent for JAVA v4.0.005 (PassNi-SSO-Agent-for-JAVA-v4.0.005.zip)
		Pass-Ni SSO Agent for ASP.NET v4.0.005 (PassNi-SSO-Agent-for-ASP.NET-v4.0.005.zip)
		Pass-Ni SSO Agent for PHP v4.0.005 (PassNi-SSO-Agent-for-PHP-v4.0.005.zip)
Electronic Documents (PDF)	Guidance Document	Pass-Ni SSO v4.0 Preparative Procedures V1.0 R5 (PassNi-SSO-v4.0-PRE-V1.0.R5.pdf)
		Pass-Ni SSO v4.0 Operational Guidance V1.0 R5 (PassNi-SSO-v4.0-OPE-V1.0.R5.pdf)

[Table 1-8] Composition of the Product and the TOE

1.4.3 The logical scope of the TOE

The logical scope of the TOE is shown in following figure.



[Figure 1-4] Logical Scope of the TOE

Security audit

The TOE Server generates and stores the audit data for the identification and authentication success / failure of the user, the TOE configuration change history, and the security function execution history. The audit data includes the date and time of the event, the type of the event, the identity of the entity that generates the event, the details of the activity and the results (success / failure), and uses the time information of the TOE installed system to generate accurate time information. The TOE Server allows the authorized administrator to review the audit data and provides the function to perform selectable audit review according to the type of audit data, time, and so on. The TOE Server also provides the function of notifying the authorized administrator through e-mail when the size of the audit trail exceeds the specified limit or when the potential security violation is detected through analysis of the audit data.

Cryptographic support

The TOE Server and the TOE agent generates a cryptographic key using the target algorithm of the evaluation in the validated cryptographic module whose security and implementation conformance has been verified through the Korean Cryptographic Module Validation Program (KCMVP). It provides the function to securely discard the cryptographic key. The TOE server and the TOE agent exchange cryptographic keys through the validated cryptographic module for cryptographic communication between the components and perform functions of symmetric key cryptography, MAC, hash and ECDH cryptographic operation to protect transmitted data and stored data. The TOE Server and the TOE agent generates a random number using the random number generator of the validated cryptographic module.

Identification and authentication

The TOE server identifies and authenticates administrators using security management functions based on ID / PW. The TOE server identifies and authenticates the end-user based on the ID / PW and issues an authentication token to the authorized end-user. When a end-user requests token-based authentication with an authentication token that has been issued, the TOE Agent verifies the authentication token from the TOE server and performs identification and authentication. The TOE Server will disable the identification and authentication function for the time set by the authorized administrator (default 5 minutes) so that the user can no longer log in if the user fails authentication more than 5 times. During user and administrator identification and authentication process, the TOE ensures that passwords input are masked and prevented from reusing user's authentication information through checking combination rules when generating and changing passwords. The TOE Sever generates an authentication token including a random number and an issuance time (timestamp) to guarantee uniqueness, and securely destroys the authentication token when the session ends or expires. TOE Server and TOE Agent perform mutual authentication through in-house implementation Handshake Authentication Protocol.

Security management

The TOE Server provides the security management function that allows the authorized administrator to set and manage security functions, security policies and important data, and rules for creating and changing ID / password. In addition, only for authorized administrator can be accessible, it provides the permission management function that can restrict the functions that can be accessed by administrator and its roles (administrator, system manager, policy manager, user manager, monitoring manager).

Protection of the TSF

The TOE Server and the TOE Agent ensure the confidentiality and integrity of transmitted data and stored TSF data The TOE Server and the TOE Agent performs self-testing periodically during start-up, during normal operation, or at the request of an authorized administrator to ensure the correct operation of each execution module, and to verify the integrity of important data such as execution codes and configurations. The TOE server communicates with the SMTP server through a secure channel and performs an external entity test to confirm that the SMTP server is operating normally when requested by the authorized administrator.

TOE access

Since establishing session of the administrator account from one terminal, the TOE Server terminates the previous session when the access is occurred by the identical account or the identical level administrator from another terminal. Each administrator session or user session is also terminated if the session is not active for a certain period of time (default: 10 minutes) after an administrator or a user login. The TOE server can block the administrator's management access according to the connection IP, whether or not to use, the start date, and the end date.

1.5 Conventions

The notation, formatting and conventions used in this ST are consistent with the Common Criteria for Information Technology Security Evaluation.

The CC allows several operations to be performed for functional requirements: iteration, assignment, selection and refinement. Each operation is used in this ST.

Iteration

Iteration is used when a component is repeated with varying operations. The result of iteration is marked with an iteration number in parenthesis following the component identifier, i.e., denoted as (iteration No.)

Assignment

This is used to assign specific values to unspecified parameters (e.g., password length). The result of assignment is indicated in square brackets like [assignment_value].

Selection

This is used to select one or more options provided by the CC in stating a requirement. The result of selection is shown as *underlined and italicized*.

Refinement

This is used to add details and thus further restrict a requirement. The result of refinement is shown in **bold text**.

“Application notes” is provided to clarify the intent of requirements, provide the information for the optional items in implementation, and define "Pass/Fail" criteria for a requirement. The application notes is provided with corresponding requirements if necessar.

1.6 Terms and Definitions

The terms used in this ST are the same as those used in the CC and PP, conform to the CC and are not further described in this ST

JWT (JSON Web Token)

It is composed of three areas (Header, Payload, Signature) in JSON expression format defined by RFC7519 standard

Link System

A business system that is installed with the SSO agent that is the TOE component and works with the SSO server

Link System ID

ID value for uniquely identifying the Link System

Link System SecretCode

A secret value to prove to the SSO server that it is a Link System

SMTP Server

A server that sends e-mail using Simple Mail Transfer Protocol (SMTP)

1.7 Security Target Contents

Chapter 1 Introduction describes the Security Target and TOE reference, TOE overview, TOE description, convention and terms and definitions

Chapter 2 Conformance Claims describes the conformance with the Common Criteria, protection profile and package and presents the conformance rationale and protection profile conformance statement.

Chapter 3 defines the security objectives for the operational environment supported by the operational environment in order to provide the security functionality of the TOE in an accurate manner.

Chapter 4 defines the extended components additionally needed according to the features of Single Sign-On.

Chapter 5 Security Requirements describes security functional requirements and security assurance requirements.

Chapter 6 describes the TOE summary specification.

2. Conformance Claim

2.1 CC Conformance Claim

This Security Target conforms to the following Common Criteria.

Common Criteria		Common Criteria for Information Technology Security Evaluation V3.1R5 - Common Criteria Part 1: Introduction and General Model V3.1r5, (CCMB-2017-04-001, 2017. 4) - Common Criteria Part 2: Security Functional Components V3.1r5, (CCMB-2017-04-002, 2017.4) - Common Criteria Part 3: Security Assurance Components V3.1r5, (CCMB-2017-04-003, 2017.4)
Conformance Claim	Part2 Security Functional Requriements	Extended: FCS_RBG.1, FIA_IMA.1, FIA_SOS.3, FMT_PWD.1, FPT_PST.1, FTA_SSL.5
	Part 3 Security Assurance Requirements	Conformant
	Package	Augmented: EAL1 augmented(ATE_FUN.1)

2.2 PP Conformance Claim

This Security Target conforms to the following Protection Profile.

- Protection Profile
  - National Protection Profile for Single Sign-On V1.0 (KECS-PP-0822-2017, 2017. 8. 18)

- PP Conformance Type
  - "Strict PP conformance"

2.3 Package Conformance Claim

This ST claims conformance to assurance requirement package EAL1 and additionally defines some assurance requirements.

- Assurance package: EAL1 augmented(ATE_FUN.1)

2.4 Conformance Claim Rationale

Since this ST adopts the TOE type, security objectives and security requirements in the same way as the Protection Profile, it is demonstrated that this ST strictly conforms to the “National Protection Profile for Single Sign-On V1.0”.

The rationale for the PP conformance claim in this Security Target is as follows.

Item	PP	ST	Conformance Rationale
Seucirty objectives	OE.PHYSIAL_CONTROL	OE.PHYSIAL_CONTROL	Same as PP
	OE.TRUSTED_ADMIN	OE.TRUSTED_ADMIN	
	OE.LOG_BACKUP	OE.LOG_BACKUP	
	OE.OPERATION_SYSTEM_REINFORCEMENT	OE.OPERATION_SYSTEM_REINFORCEMENT	
	OE.SECURE_DEVELOPMENT	OE.SECURE_DEVELOPMENT	
	OE.AUTHENTICATION_SYSTEM_SECURITY	-	Exclude from PP - In the initial authentication stage of the TOE, identification and authentication functions of endusers are not upported by external authentication systems nd therefore, the security goal of ‘OE.AUTHENTICATION_SYSTEM_SECURITY’ does not correspond.
	-	OE.TRUSTED_TIMESTAMP	More restrictive than PP - This ST is further defined in this ST

	-	OE.TRUSTED_SMTP	for security objectives that must be addressed by the technical / procedural means supported by the operational environment to provide security functionality. Therefore, this ST is more restrictive than PP because it defines 'the TOE operating environment to deal with these additional security objectives'
	-	OE.TRUSTED_AUDIT_STORAGE	
	-	OE.SECURE_CHANNEL	

3. Security objectives

The followings are the security objectives handled by technical and procedural method supported from operational environment in order to provide the TOE security functionality accurately.

3.1 Security objectives for the operational environment

This ST conforms to the security objectives for all operating environments specified in the PP. The following are security objectives to be addressed by the technical / procedural means supported by the operational environment so that the TOE can accurately provide security functionality.

- **OE.PHYSICAL_CONTROL**

The place where SSO agent and SSO server among the TOE components are installed and operated shall be equipped with access control and protection facilities so that only authorized administrator can access.

- **OE.TRUSTED_ADMIN**

The authorized administrator of the TOE shall be non-malicious users, have appropriately trained for the TOE management functions and accurately fulfill the duties in accordance with administrator guidances.

- **OE.LOG_BACKUP**

The authorized administrator shall periodically checks a spare space of audit data storage in case of the audit data loss, and carries out the audit data backup (external log server or separate storage device, etc.) to prevent audit data loss.

- **OE.OPERATION_SYSTEM_REINFORCEMENT**

The authorized administrator of the TOE shall ensure the reliability and security of the operating system by performing the reinforcement on the latest vulnerabilities of the operating system in which the TOE is installed and operated.

Application notes

○ Depending on the implementation type of the TOE, the TOE components(SSO agent, SSO server) may not use the operating system independently, so care shall be taken that the operating system related settings of other external entities operating in the same operating system do not affect the secure operation of the TOE.

- **OE.SECURE_DEVELOPMENT**

The developer who uses the TOE to interoperate with the user identification and authentication function in the operational environment of the business system shall ensure that the security functions of the TOE are securely applied in accordance with the requirements of the manual provided with the TOE.

- **OE.SECURE_CHANNEL**

The Web Application Server (WAS) that interfaces with the TOE provides a secure and trusted channel so that all information transmitted when the user accesses through the web browser should be securely protected.

- **OE.TRUSTED_TIMESTAMP**

The TOE should be used with reliable time information provided by TOE operating environment.

- **OE.TRUSTED_SMTP**

The authorized administrator of the TOE shall set secure and reliable SMTP server information so that the TOE can send mail to the secure path when installing the TOE.

- **OE.TRUSTED_AUDIT_STORAGE**

The audit storage associated with the TOE must ensure that it maintains secure and trusted operations.

4. Extended components definition

4.1 Cryptographic support

4.1.1 Random Bit Generation

Family Behaviour

This family defines requirements for the TSF to provide the capability that generates random bits required for TOE cryptographic operation.

Component leveling

FCS_RBG Random bit generation
1

FCS_RBG.1 random bit generation, requires TSF to provide the capability that generates random bits required for TOE cryptographic operation.

Management: FCS_RBG.1

There are no management activities foreseen.

Audit: FCS_RBG.1

There are no auditable events foreseen.

4.1.1.1 FCS_RBG.1 Random bit generation

Hierarchical to	No other components
Dependencies	No dependencies.

FCS_RBG.1.1 The TSF shall generate random bits required to generate an cryptographic key using the specified random bit generator that meets the following [assignment: *list of standards*].

4.2 Identification and authentication

4.2.1 TOE Internal mutual authentication

Family Behaviour

This family defines requirements for providing mutual authentication between TOE components in the process of user identification and authentication

Component leveling

FIA_IMA TOE Internal mutual authentication
1

FIA_IMA.1 TOE Internal mutual authentication requires that the TSF provides mutual authentication function between TOE components in the process of user identification and authentication.

Management: FIA_IMA.1

There are no management activities foreseen.

Audit: FIA_IMA.1

The following actions are recommended to record if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimum: Success and failure of mutual authentication

4.2.1.1 FIA_IMA.1 TOE Internal mutual authentication

Hierarchical to No other components

Dependencies No dependencies.

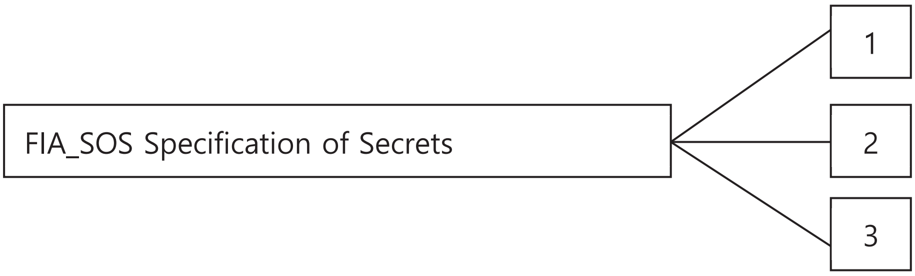
FIA_IMA.1.1 The TSF shall perform mutual authentication between [assignment: *different parts of TOE*] using the [assignment: authentication protocol] that meets the following [assignment: *list of standards*].

4.2.2 Specification of Secrets

Family Behaviour

This family defines requirements for mechanisms that enforce defined quality metrics on provided secrets and generate secrets to satisfy the defined metric.

Component leveling



The specification of secrets family in CC Part 2 is composed of 2 components. It is now composed of three components, since this PP adds one more component as below.

※ The description on two components included in CC Part 2 is omitted.

FIA_SOS.3 Destruction of secrets requires, that the secret information be destroyed according to the specified destruction method, which can be based on the assigned standard.

Management: FIA_SOS.3

There are no management activities foreseen

Audit: FIA_SOS.3

The following actions are recommended to record if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimum : Success and failure of the activity.

4.2.2.1 FIA_SOS.3 Destruction of Secrets

Hierarchical to	No other components
Dependencies	FIA_SOS.2 TSF Generation of secrets.

FIA_SOS.3.1 The TSF shall destroy secrets in accordance with a specified secrets destruction method [assignment: *secret destruction method*] that meets the following: [assignment: *list of standards*].

Application notes

○ This SFR can be applied to the user's token.

4.3 Security Management

4.3.1 ID and password

Family Behaviour

This family defines the capability that is required to control ID and password management used in the TOE, and set or modify ID and/or password by authorized users.

Component leveling

FMT_PWD ID and password
1

FMT_PWD.1 ID and password management, requires that the TSF provides the management function of ID and password

Management: FMT_PWD.1

The following actions could be considered for the management functions in FMT:.

a) Management of ID and password configuration rules.

Audit: FMT_PWD.1

The following actions are recommended to record if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimum: All changes of the password

4.3.1.1 FMT_PWD.1 Management of ID and password

Hierarchical to	No other components
Dependencies	FMT_SMF.1 Specification of Management Functions FMT_SMR.1 Security roles

FMT_PWD.1.1 The TSF shall restrict the ability to manage the password of [assignment: *list of functions*] to [assignment: the authorized identified roles].

1. [assignment: *password combination rules and/or length*]

2. [assignment: *other management such as management of special characters unusable for password, etc.*]

FMT_PWD.1.2 The TSF shall restrict the ability to manage the ID of [assignment: *list of functions*] to [assignment: *the authorized identified roles*].

1. [assignment: *ID combination rules and/or length*]

2. [assignment: *other management such as management of special characters unusable for ID, etc.*]

FMT_PWD.1.3 The TSF shall provide the capability for [selection, choose one of: *setting ID and password when installing, setting password when installing, changing the ID and password when the authorized administrator accesses for the first time, changing the password when the authorized administrator accesses for the first time*].

Application notes

○ If the TOE does not provide the capability for managing the ID and password combination rules by authorized roles, etc., ‘None.’ may be specified in assignment operations of FMT_PWD.1.1, FMT_PWD.1.2.

○ The ID and password combination rules that can be set by authorized roles may include minimum and maximum length setting, mixing rule setting involving English upper case/lower case/number/special characters, etc.

4.4 Protection of the TSF

4.4.1 Protection of stored TSF data

Family Behaviour

This family defines rules to protect TSF data stored within containers controlled by the TSF from the unauthorized modification or disclosure.

Component leveling

FPT_PST Protection of stored TSF data
1

FPT_PST.1 Basic protection of stored TSF data, requires the protection of TSF data stored in containers controlled by the TSF.

Management: FPT_PST.1

There are no management activities foreseen.

Audit: FPT_PST.1

There are no auditable events foreseen

4.4.1.1 FPT_PST.1 Basic protection of stored TSF data

Hierarchical to No other components

Dependencies No dependencies.

FPT_PST.1.1 The TSF shall protect [assignment: *TSF data*] stored in containers controlled by the TSF from the unauthorized [selection: *disclosure, modification*].

Application notes

○ Containers controlled by the TSF mean storage in the TOE or external entities (DBMS, etc.) that interact with the TOE.

○ Examples of TSF data to be protected as follows:

- User password, cryptographic key (pre-shared key, symmetric key, private key, etc), TOE configuration values (security policy, environment setting, configuration parameters), audit data, etc.

○ The TSF data can be encrypted and stored to be protected from the unauthorized disclosure or modification.

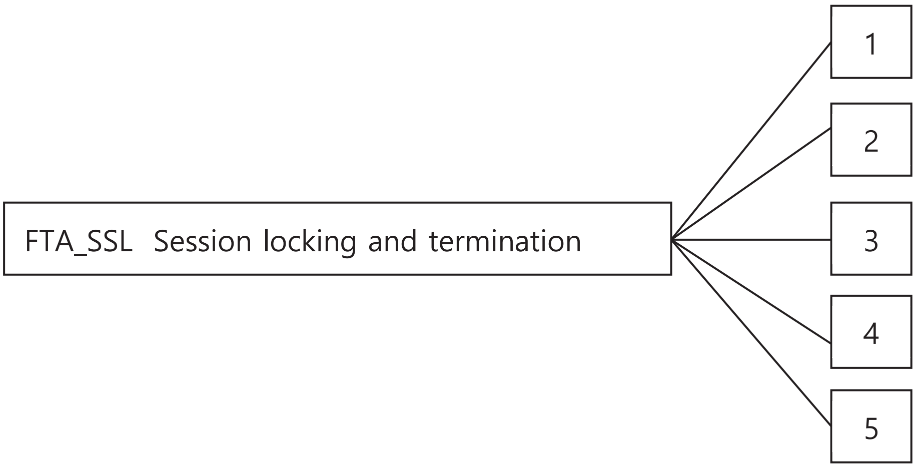
4.5 TOE Access

4.5.1 Session locking and termination

Family Behaviour

This family defines requirements for the TSF to provide the capability for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

Component leveling



In CC Part 2, the session locking and termination family consists of four components. In this PP, it consists of five components by extending one additional component as follows.

※ The relevant description for four components contained in CC Part 2 is omitted.

FTA_SSL.5 The management of TSF-initiated sessions, provides requirements that the TSF locks or terminates the session after a specified time interval of user inactivity.

Management: FTA_SSL.5

The following actions could be considered for the management functions in FMT:

a) Specification for the time interval of user inactivity that is occurred the session locking and termination for each user

b) Specification for the time interval of default user inactivity that is occurred the session locking and termination

Audit: FTA_SSL.5

The following actions are recommended to record if FAU_GEN Security audit data generation is included in the PP/ST:.

a) Minimum: Locking or termination of interactive session

4.5.1.1 FTA_SSL.5 Management of TSF-initiated sessions

Hierarchical to	No other components.
Dependencies	FIA_UAU.1 Authentication or No dependencies.

FTA_SSL.5.1 The TSF shall [selection:

• *lock the session and re-authenticate the user before unlocking the session,*

• *terminate] an interactive session after a [assignment: time interval of user inactivity].*

Application notes

○ This requirement can be applied to the management access of user(SSH, HTTPS, etc.).

5. Security requirements

The security requirements describe security functional requirements and assurance requirements that must be satisfied by the TOE that claims conformance to this ST.

5.1 Security functional requirements

The following table summarizes the security functional requirements used in the ST.

Security Functional Class	Security Functional Component		Remarks
Security Audit (FAU)	FAU_ARP.1	Security alarms	
	FAU_GEN.1	Audit data generation	
	FAU_SAA.1	Potential violation analysis	
	FAU_SAR.1	Audit review	
	FAU_SAR.3	Selectable audit review	
	FAU_STG.3	Action in case of possible audit data loss	
	FAU_STG.4	Prevention of audit data loss	
Cryptographic Support (FCS)	FCS_CKM.1	Cryptographic key generation	
	FCS_CKM.2	Cryptographic key distribution	
	FCS_CKM.4	Cryptographic key destruction	
	FCS_COP.1(1)	Cryptographic operation (Symmetric key cryptographic operation)	
	FCS_COP.1(2)	Cryptographic operation (MAC)	
	FCS_COP.1(3)	Cryptographic operation (Hash)	
	FCS_COP.1(4)	Cryptographic operation (ECDH)	
	FCS_RBG.1(Extended)	Random bit generation	
Identification and Authentication (FIA)	FIA_AFL.1(1)	Authentication failure handling (End-user)	
	FIA_AFL.1(2)	Authentication failure handling (Administrator)	
	FIA_IMA.1(Extended)	TOE Internal mutual authentication	
	FIA_SOS.1	Verification of secrets	
	FIA_SOS.2	TSF Generation of secrets	
	FIA_SOS.3(Extended)	Destruction of secrets	
	FIA_UAU.1	Timing of authentication	

	FIA_UAU.4	Single-use authentication mechanisms	
	FIA_UAU.7	Protected authentication feedback	
	FIA_UID.1	Timing of identification	
Security Management (FMT)	FMT_MOF.1	Management of security functions behaviour	
	FMT_MTD.1	Management of TSF data	
	FMT_PWD.1(Extended)	Management of ID and password	
	FMT_SMF.1	Specification of management functions	
	FMT_SMR.1	Security roles	
Protection of the TSF (FPT)	FPT_ITT.1	Basic internal TSF data transfer protection	
	FPT_PST.1(Extended)	Basic protection of stored TSF data	
	FPT_TST.1	TSF testing	
	FPT_TEE.1	Testing of external entities	Optional SFR
TOE Access (FTA)	FTA_MCS.2	Per user attribute Limitation on multiple concurrent sessions	
	FTA_SSL.5(Extended)	Management of TSF-initiated sessions	
	FTA_TSE.1	TOE session establishment	
Trusted path/channels (FTP)	FTP_ITC.1	Inter-TSF trusted channel	Optional SFR

[Table 5-1] Security functional requirements (SFR)

5.1.1 Security audit (FAU)

5.1.1.1 FAU_ARP.1 Security alarms

Hierarchical to No other components.

Dependencies FAU_SAA.1 Potential violation analysis

FAU_ARP.1.1 The TSF shall take [Sending mail to the e-mail address specified by the authorized administrator] upon detection of a potential security violation.

5.1.1.2 FAU_GEN.1 Audit data generation

Hierarchical to No other components.

Dependencies FPT_STM.1 Reliable time stamps

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) [Refer to the "auditable events" in **[Table 5-2]** Audit events, [N/A]]

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST [Refer to the contents of "additional audit record" in **[Table 5-2]** Audit events, [N/A]]

Security functional component	Auditable event	Additional audit record
FAU_ARP.1	Actions taken due to potential security violations	
FAU_SAA.1	Enabling and disabling of any of the analysis mechanisms, Automated responses performed by the tool	
FAU_STG.3	Actions taken due to exceeding of a threshold	
FAU_STG.4	Actions taken due to the audit storage failure	
FCS_CKM.1	Success and failure of the activity	
FCS_CKM.2	Success and failure of the activity (only applying to key distribution related to the TSF data encryption/decryption)	
FCS_CKM.4	Success and failure of the activity (only applying to key destruction related to the TSF data encryption/decryption)	
FCS_COP.1	Success and failure, and the type of cryptographic operation(only applying to items related to the issue, storing, verification, and destruction of a token)	
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts and the actions taken, and the subsequent, if appropriate, restoration to the normal state	

FIA_IMA.1(Extended)	Success and failure of mutual authentication	
FIA_SOS.2	Rejection by the TSF of any tested secret	
FIA_SOS.3(확장)	Success and failure of the activity(applicable to the destruction of SSO token only)	
FIA_UAU.1	All use of the authentication mechanism	
FIA_UAU.4	Attempts to reuse authentication data	
FIA_UID.1	All use of the administrator identification mechanism, including the administrator identity provided	
FMT_MOF.1	All modifications in the behaviour of the functions in the TSF	
FMT_MTD.1	All modifications to the values of TSF data	Modified values of TSF data
FMT_PWD.1(Extended)	All changes of the password	
FMT_SMF.1	Use of the management functions	
FMT_SMR.1	Modifications to the user group of rules divided	
FPT_TST.1	Execution of the TSF self tests and the results of the tests	Modified TSF data or execution code in case of integrity violation
FPT_TEE.1	Success and failure of the activity	
FTA_MCS.2	Denial of a new session based on the limitation of multiple concurrent sessions	
FTA_SSL.5(Extended)	Locking or termination of interactive session	
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism All attempts at establishment of a user session	
FTP_ITC.1	Failure of the trusted channel functions Identification of the initiator and target of failed trusted channel functions	

[Table 5-2] Audit events

5.1.1.3 FAU_SAA.1 Potential violation analysis

Hierarchical to No other components

Dependencies FAU_GEN.1 Audit data generation

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

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

a) Accumulation or combination of [
An auditable event of authentication failure in FIA_UAU.1,
An auditable event of integrity violation in FPT_TST.1,
self-test failure of the validated cryptographic module,
Audit trail exceeding 80 percent of the threshold,
Audit safe fails
] known to indicate a potential security violation;

b) N/A]

5.1.1.4 FAU_SAR.1 Audit review

Hierarchical to No other components.
Dependencies FAU_GEN.1 Audit data generation

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

5.1.1.5 FAU_SAR.3 Selectable audit review

Hierarchical to No other components
Dependencies FAU_SAR.1 Audit review

FAU_SAR.3.1 The TSF shall provide the ability to apply [the following methods of selection and/or ordering] of audit data based on [the following criteria with logical relations].

[

- Criteria with logical relations: [Table 5-3] Audit Data Type and Selection Criteria
- Methods of selection and/or ordering: Ordering in the descending order based on the number of audit data log generation or Ordering

in descending/ascending order according to column name in search result

]

Audit Data Type	Selection Criteria	Allowable Ability
System Audit	(General search) - Search period: start date ~ end date AND - Type: keyword OR - Type name: keyword OR - Result code: keyword (Detailed search) - No: keyword, equal sign/inequality sign/pattern, AND/OR - State: keyword, equal sign/inequality sign/pattern, AND/OR - Type: keyword, equal sign/inequality sign/pattern, AND/OR - Type name: keyword, equal sign/inequality sign/pattern, AND/OR - Result code: keyword, equal sign/inequality sign/pattern, AND/OR - Message: keyword, equal sign/inequality sign/pattern, AND/OR - Date: Select date, equal sign/inequality sign/pattern, AND/OR - Remarks: Select date, equal sign/inequality sign/pattern, AND/OR	Search, Sort, View details
Business System Audit	(General search) - Search period: start date ~ end date AND - Link system ID: keyword OR - Link system name: keyword OR - Type: keyword OR - Type name: keyword OR - Result code: keyword (Detailed search) - No: keyword, equal sign/inequality sign/pattern, AND/OR - Link system ID: keyword, equal sign/inequality sign/pattern, AND/OR - Link system name: keyword, equal sign/inequality sign/pattern, AND/OR - State: keyword, equal sign/inequality sign/pattern, AND/OR - Type: keyword, equal sign/inequality sign/pattern, AND/OR - Type name: keyword, equal sign/inequality sign/pattern, AND/OR - Result code: keyword, equal sign/inequality sign/pattern, AND/OR - Message: keyword, equal sign/inequality sign/pattern, AND/OR - Date: Select date, equal sign/inequality sign/pattern, AND/OR - Parameter: keyword, equal sign/inequality sign/pattern, AND/OR - User ID: keyword, equal sign/inequality sign/pattern, AND/OR - User name: keyword, equal sign/inequality sign/pattern	Search, Sort, View details

Administrator access	(General search) - Search period: start date ~ end date AND - ID: keyword OR - Name: keyword OR - IP: keyword (Detailed search) - No: keyword, equal sign/inequality sign/pattern, AND/OR - ID: keyword, equal sign/inequality sign/pattern, AND/OR - Name: keyword, equal sign/inequality sign/pattern, AND/OR - IP: keyword, equal sign/inequality sign/pattern, AND/OR - Device: keyword, equal sign/inequality sign/pattern, AND/OR - OS: keyword, equal sign/inequality sign/pattern, AND/OR - Browser: keyword, equal sign/inequality sign/pattern, AND/OR - Browser version: keyword, equal sign/inequality sign/pattern, AND/OR - State: keyword, equal sign/inequality sign/pattern, AND/OR - Type: keyword, equal sign/inequality sign/pattern, AND/OR - Type name: keyword, equal sign/inequality sign/pattern, AND/OR - Result code: keyword, equal sign/inequality sign/pattern, AND/OR - Message: keyword, equal sign/inequality sign/pattern, AND/OR - Login date: Select date, equal sign/inequality sign/pattern, AND/OR - Logout date: Select date, equal sign/inequality sign/pattern, AND/OR - Whether to destructed the authentication token: keyword, equal sign/inequality sign/pattern	Search, Sort, View details
Administrator activity	(General search) - Search period: start date ~ end date AND - ID: keyword OR - Name: keyword OR - IP: keyword (Detailed search) - No: keyword, equal sign/inequality sign/pattern, AND/OR - ID: keyword, equal sign/inequality sign/pattern, AND/OR - Name: keyword, equal sign/inequality sign/pattern, AND/OR - IP: keyword, equal sign/inequality sign/pattern, AND/OR - Device: keyword, equal sign/inequality sign/pattern, AND/OR - OS: keyword, equal sign/inequality sign/pattern, AND/OR - Browser: keyword, equal sign/inequality sign/pattern, AND/OR - Browser version: keyword, equal sign/inequality sign/pattern, AND/OR - State: keyword, equal sign/inequality sign/pattern, AND/OR - Type: keyword, equal sign/inequality sign/pattern, AND/OR - Type name: keyword, equal sign/inequality sign/pattern, AND/OR	Search, Sort, View details

	- Result code: keyword, equal sign/inequality sign/pattern, AND/OR - Message: keyword, equal sign/inequality sign/pattern, AND/OR - Date: Select date, equal sign/inequality sign/pattern, AND/OR - Access URI: keyword, equal sign/inequality sign/pattern, AND/OR - Resource name: keyword, equal sign/inequality sign/pattern, AND/OR - Parameter: keyword, equal sign/inequality sign/pattern	
End-user access	(General search) - Search period: start date ~ end date AND - ID: keyword OR - Name: keyword OR - IP: keyword (Detailed search) - No: keyword, equal sign/inequality sign/pattern, AND/OR - ID: keyword, equal sign/inequality sign/pattern, AND/OR - Name: keyword, equal sign/inequality sign/pattern, AND/OR - IP: keyword, equal sign/inequality sign/pattern, AND/OR - Device: keyword, equal sign/inequality sign/pattern, AND/OR - OS: keyword, equal sign/inequality sign/pattern, AND/OR - Browser: keyword, equal sign/inequality sign/pattern, AND/OR - Browser version: keyword, equal sign/inequality sign/pattern, AND/OR - State: keyword, equal sign/inequality sign/pattern, AND/OR - Type: keyword, equal sign/inequality sign/pattern, AND/OR - Type name: keyword, equal sign/inequality sign/pattern, AND/OR - Result code: keyword, equal sign/inequality sign/pattern, AND/OR - Message: keyword, equal sign/inequality sign/pattern, AND/OR - Login type: keyword, equal sign/inequality sign/pattern, AND/OR - Login typename: keyword, equal sign/inequality sign/pattern, AND/OR - Login date: Select date, equal sign/inequality sign/pattern, AND/OR - Login link system id: keyword, equal sign/inequality sign/pattern, AND/OR - Login link system name: keyword, equal sign/inequality sign/pattern, AND/OR - Logout date: Select date, equal sign/inequality sign/pattern, AND/OR - Logout link system id: keyword, equal sign/inequality sign/pattern, AND/OR - Logout link system name: keyword, equal sign/inequality sign/pattern, AND/OR - Whether to destructed the authentication token: keyword, equal sign/inequality sign/pattern, AND/OR	Search, Sort, View details
End-user	(General search)	Search,

activity	- Search period: start date ~ end date AND - ID: keyword OR - Name: keyword OR - IP: keyword (Detailed search) - No: keyword, equal sign/inequality sign/pattern, AND/OR - ID: keyword, equal sign/inequality sign/pattern, AND/OR - Name: keyword, equal sign/inequality sign/pattern, AND/OR - IP: keyword, equal sign/inequality sign/pattern, AND/OR - Device: keyword, equal sign/inequality sign/pattern, AND/OR - OS: keyword, equal sign/inequality sign/pattern, AND/OR - Browser: keyword, equal sign/inequality sign/pattern, AND/OR - Browser version: keyword, equal sign/inequality sign/pattern, AND/OR - State: keyword, equal sign/inequality sign/pattern, AND/OR - Type: keyword, equal sign/inequality sign/pattern, AND/OR - Type name: keyword, equal sign/inequality sign/pattern, AND/OR - Result code: keyword, equal sign/inequality sign/pattern, AND/OR - Message: keyword, equal sign/inequality sign/pattern, AND/OR - Date: Select date, equal sign/inequality sign/pattern, AND/OR - Access URI: keyword, equal sign/inequality sign/pattern, AND/OR - Resource name: keyword, equal sign/inequality sign/pattern, AND/OR - Parameter: keyword, equal sign/inequality sign/pattern	Sort, View details
※ Legend: equal sign(=), inequality sign (<>, >, <, >=, <=), pattern(LIKE)		

[Table 5-3] Audit Data Type and Selection Criteria

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

Hierarchical to No other components

Dependencies FAU_GTG.1 Protected audit trail storage

FAU_STG.3.1 The TSF shall [Notification to the authorized administrator, [N/A] if the audit trail exceeds [80% of database's volume capacity (fixed value)]].

5.1.1.7 FAU_STG.4 Prevention of audit data loss

Hierarchical to FAU_STG.3 Action in case of possible audit data loss

Dependencies FAU_GTG.1 Protected audit trail storage

FAU_STG.4.1 The TSF shall "*overwrite the oldest stored audit records*" and [Sending mail to the e-mail address specified by the authorized administrator] if the audit trail is full.

5.1.2 Cryptographic support (FCS)

5.1.2.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to No other components

Dependencies [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ["cryptographic key generation algorithm" in the following table] and specified cryptographic key sizes ["cryptographic key sizes" in the following table] that meet the following: ["list of standards" in the following table].

Item	Cryptographic Key generation Algorithm	Detail	cryptographic Key Sizes (bits)	List of Standards
KEK	PBKDF2	HMAC-SHA256 Salt: 128bits Iteration: 1024	128	TTAK.KO-12.0274
Master Key	Hash_DRBG	Hash: SHA-256	Symmetric key: 256 MAC: 160	ISO/IEC 18031
Secret Key	Hash_DRBG	Hash: SHA-256	128	ISO/IEC 18031
Elliptic curve key pair	Hash_DRBG	Hash: SHA-256 Curve: P-256	256	ISO/IEC 18031
Shared Key	ECDH	Hash: SHA-256 Curve: P-256	Symmetric key: 256 MAC: 160	ISO/IEC 11770-3

[Table 5-4] Cryptographic key generation algorithm

Application Notes

○ The cryptographic algorithm and cryptographic key sizes shall meet the cryptographic complexity of 112 bits or more.

○ Generating a cryptographic key by deriving it from the password is not allowed, except the key encryption key (KEK).

5.1.2.2 FCS_CKM.2 Cryptographic key distribution

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [ECDH (Elliptic Curve Diffie-Hellman) elliptic-curve based key establishment mechanisms (Curve: P-256)] that meets the following: [ISO/IEC 11770-3].

5.1.2.3 FCS_CKM.4 Cryptographic key destruction

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [How to overwrite all of the digits with zeros] that meets the following: [Cryptographic key destruction method specified in the Cryptographic Key Management Guide (Ministry of Science, ICT and Future Planning, 2014)]

5.1.2.4 FCS_COP.1(1) Cryptographic operation (Symmetric key cryptographic operation)

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(1) The TSF shall perform [“list of cryptographic operations” in the following table] in accordance with a specified cryptographic algorithm [“cryptographic algorithm” in the following table] and cryptographic key sizes [“cryptographic key sizes” in the following table] that meet the following: [“list of standards” in the following table]

Cryptographic alogrithm	Key type	Cryptographic key sizes(bits)	Operation Mode	List of standards	List of cryptographic operations
SEED	KEK	128	CBC	TTAS.KO-12.0004/R1	- private key of the TOE server and TOE agent
ARIA	Master Key	256	CBC	KS X 1213-1	- environment configuration file - Security policy such as ID / PW setting rule - authentication token
	Shared Key	256	CBC	KS X 1213-1	- Transfer data between TOE Server and Agent

[Table 5-5] List of cryptographic operation (Symmetric key)

5.1.2.5 FCS_COP.1(2) Cryptographic operation (MAC)

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(2) The TSF shall perform [“list of cryptographic operations” in the following table] in accordance with a specified cryptographic algorithm [“cryptographic algorithm” in the following table] and cryptographic key sizes [“cryptographic key sizes” in the following table] that meet the following: [“list of standards” in the following table]

Cryptographic alogrithm	Key type	Cryptographic key sizes(bits)	Hash	List of standards	List of cryptographic operations

HMAC	Master Key	160	SHA-256	ISO/IEC 9797-2	- Integrity verification of the authentication token
	Shared Key	160	SHA-256	ISO/IEC 9797-2	- Integrity verification of transfer data between TOE Server and Agent
	Secret Key	128	SHA-256	ISO/IEC 9797-2	- TOE Internal mutual authentication

[Table 5-6] List of cryptographic operation (MAC)

5.1.2.6 FCS_COP.1(3) Cryptographic operation (Hash)

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(3) The TSF shall perform ["list of cryptographic operations" in the following table] in accordance with a specified cryptographic algorithm ["cryptographic algorithm" in the following table] and cryptographic key sizes ["cryptographic complexity" in the following table] that meet the following: ["list of standards" in the following table]

Cryptographic alogrithm	Hash	cryptographic complexity[1] (bits)	List of standards	List of cryptographic operations
HASH	SHA-256	128	ISO/IEC 10118-3	- End-user/administrator's password - TOE's Integrity verification code

[Table 5-7] List of cryptographic operation (Hash)

5.1.2.7 FCS_COP.1(4) Cryptographic operation (ECDH)

Hierarchical to No other components

Dependencies [FDP_ITC.1 Import of user data without security attributes, or

[1] The hash cryptographic operation does not use a cryptographic key, so it is replaced by a cryptographic complexity(security strength)

FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(4) The TSF shall perform ["list of cryptographic operations" in the following table] in accordance with a specified cryptographic algorithm ["cryptographic algorithm" in the following table] and cryptographic key sizes ["cryptographic key sizes" in the following table] that meet the following: ["list of standards" in the following table]

Cryptographic alogrithm	Key type	Curve (Hash)	Cryptographic key sizes(bits)	List of standards	List of cryptographic operations
ECDH	Elliptic curve key pair	P-256 (SHA-256)	256	ISO/IEC 11770-3	- Distributing cryptographic keys between TOE Server and TOE Agent(Shared Key)

[Table 5-8] List of cryptographic operation (ECDH)

5.1.2.8 FCS_RBG.1 Random bit generation (Extended)

Hierarchical to No other components
Dependencies No dependencies.

FCS_RBG.1.1 The TSF shall generate random bits required to generate an cryptographic key using the specified random bit generator that meets the following [ISO/IEC 18031 Hash_DRBG(SHA-256)].

5.1.3 Identification and authentication (FIA)

5.1.3.1 FIA_AFL.1(1) Authentication failure handling (End-user)

Hierarchical to No other components
Dependencies FIA_UAU.1 Timing of authentication

FIA_AFL.1.1(1) The TSF shall detect when *an administrator configurable positive integer within [3 ~ 10 (default value 5)]* unsuccessful authentication attempts occur related to [authentication attempt of end-user].

FIA_AFL.1.2(1) When the defined number of unsuccessful authentication attempts has been *met* the TSF shall [the following list of actions

a) Disable the user identification and authentication function during a positive number of minutes (5 to 60) configurable by the administrator (temporary blocking, default value 5 minutes) or

b) Disable (block) the identification and authentication functions until the administrator unlocks disabled identification and authentication functions for the user

].

5.1.3.2 FIA_AFL.1(2) Authentication failure handling (Administrator)

Hierarchical to No other components

Dependencies FIA_UAU.1 Timing of authentication

FIA_AFL.1.1(2) The TSF shall detect when *an administrator configurable positive integer within [3 ~ 10 (default value 5)]* unsuccessful authentication attempts occur related to [authentication attempt of administrator].

FIA_AFL.1.2(2) When the defined number of unsuccessful authentication attempts has been *met* the TSF shall [the following list of actions

a) Disable the user identification and authentication function during a positive number of minutes (5 to 60) configurable by the administrator (temporary blocking, default value 5 minutes) or

b) Disable (block) the identification and authentication functions until the administrator unlocks the disabled identification and authentication functions for the user].

Application notes

○ The top administrator who is provided at the time of TOE installation disables (temporary blocking) the identification and authentication function for 5 minutes regardless of the administrator's action when the number of authentication failures is reached.

5.1.3.3 FIA_IMA.1 TOE Internal mutual authentication

Hierarchical to No other components

Dependencies No dependencies.

FIA_IMA.1.1 The TSF shall perform mutual authentication between [Pass-Ni SSO Server, Pass-Ni SSO Agent] using the [In-house implementation Handshake Authentication Protocol] that meets the following [N/A]

5.1.3.4 FIA_SOS.1 Verification of secrets

Hierarchical to No other components

Dependencies No dependencies.

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [Administrator-defined permission criteria in FMT_PWD.1].

Application notes

○ The information that shall meet password complexity requirements can be data as the following

- administrator's password, end-user's password

5.1.3.5 FIA_SOS.2 TSF Generation of secrets

Hierarchical to No other components

Dependencies No dependencies.

FIA_SOS.2.1 TSF shall provide a mechanism to generate **an authentication token** that meet [the following acceptable standard

a) The subject of token generation and authentication is the SSO server.

b) The authentication token contains components as [Table 5-9].

c) The header and data contained in the authentication token must ensure integrity.

d) The data contained in the authentication token must be encrypted with a validated cryptographic module to provide confidentiality.

e) Among the authentication token components, 'subject (user)' information must ensure uniqueness and generated using a random number generator of a validated cryptographic module.

]

FIA_SOS.2.2 TSF shall be able to enforce the use of TSF-generated **authentication token** for [End-user identification and authentication].

Type	Composition	Description	subject of generation
Header	Algorithm	Signature Alogorithm(default: HMAC-SHA256)	SSO Server
Payload	Issuser	SSO server ID that issued the authentication token	
	Subject(user)	Session ID to identify the user	
	IP	User access ip	
	TIme of issue	Time that the authentication token was issued	
	Time of expiration	Expiration time of authentication token	
Signature	Integrity verification code	Header and data integrity verification code with algorithm defined in header	

[Table 5-9] Structure of authentication token

5.1.3.6 FIA_SOS.3 Destruction of secrets (Extended)

Hierarchical to No other components

Dependencies FIA_SOS.2 TSF Generation of secrets.

FIA_SOS.3.1 The TSF shall destroy **authentication tokens** in accordance with a specified **authentication token** destruction method [Overwrite with null] that meets the following: [N/A]

5.1.3.7 FIA_UAU.1 Timing of authentication

Hierarchical to No other components

Dependencies FIA_UID.1 Timing of identification

FIA_UAU.1.1 The TSF shall allow [the following list of TSF mediated actions
- Generate public key pair for ECDH key exchange
- Generate a nonce value to prevent replay attack
- Request identification and authentication procedure (login user interface display)

] 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.

5.1.3.8 FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to	No other components
Dependencies	No dependencies.

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [the following identified authentication mechanism(s)

a) Password based authentication (end-user, administrator)

b) Authentication token based authentication (end-user)].

Application notes

○ This SFR defines the requirements for the authentication data and token of the authorized administrator and the authorized end-user.

5.1.3.9 FIA_UAU.7 Protected authentication feedback

Hierarchical to	No other components
Dependencies	FIA_UAU.1 Timing of authentication.

FIA_UAU.7.1 The TSF shall provide only [the following list of feedback

a) Passwords being entered are masked (password masking with "● ● ● ● ●") during administrator / user password creation, change and administrator / user authentication

b) When administrator identification and authentication fails, the following message

- The id or password is incorrect.

c) When end-user identification and authentication fails, the following message

- The id or password is incorrect.

] to the user while the authentication is in progress.

5.1.3.10 FIA_UID.1 Timing of identification

Hierarchical to	No other components
Dependencies	No dependencies.

FIA_UID.1.1 The TSF shall allow [the following list of TSF-mediated actions

- Generate public key pair for ECDH key exchange
- Generate a nonce value to prevent replay attack
- Request identification and authentication procedure (login user interface display)

] 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.

5.1.4 Security management (FMT)

5.1.4.1 FMT_MOF.1 Management of security functions behaviour

Hierarchical to No other components

Dependencies FMT_SMF.1 Specification of Management Functions
FMT_SMR.1 Security roles.

FMT_MOF.1.1 The TSF shall restrict the ability to ***conduct management actions of*** the functions [list of functions in the following table] to [**the authorized administrator in the following table**].

List of functions	Management actions					Administrator
	determine the behavior	disable	enable	modify the behaviour of	remarks	
Management of id configuration rules	-	○	○	○		top administrator, policy administrator
Management of password configuration rules	-	○	○	○		
Management of actions to be taken in the event of an authentication failure(end-user)	-	○	○	○		
Management of	-	-	-	○		

actions to be taken in the event of an authentication failure(administrator)						
Actions to be taken when authentication is attempted with password change period exceeded (end-user)	-	○	○	○		
Actions to be taken when authentication is attempted with password change period exceeded (administrator)	-	-	-	○		
Actions to be taken in event of inactivity timeout(end-user)	-	○	○	○	Administrators have fixed values	
Management of actions to be taken in the event of a potential violation	-	○	○	-	Whether to send mail	
Management of server self-tests	-	-	-	○	Execution time	top administrator, system administrator
Management of agent self-tests	-	-	-	○	Execution time	

[Table 5-10] List of security management functions

Application notes

○ "Management action" to which a refinement operation is applied includes the ability to determine the behavior, disable, enable, modify the behavior of some functions in the TSF. This requirement shall be applied to the management access(SSH, HTTPS, etc.) supported by the TOE

○ The action that adds, deletes or modifies conditions or rules capable of determining the security functions behavior is included in the management of security functions behaviors. And, the action that adds, deletes or modifies behaviors taken by the TSF according to the corresponding conditions and rules is also included in the management of security functions behaviors. In addition, the action of selecting mechanism, protocol, etc., when there are variously provided to support the same purpose, is included in the management of security functions behavior because it corresponds to the modification of behavior.

5.1.4.2 FMT_MTD.1 Management of TSF data

Hierarchical to No other components

Dependencies FMT_SMF.1 Specification of Management Functions
FMT_SMR.1 Security roles.

FMT_MTD.1.1 The TSF shall restrict the ability to manage the [list of TSF data in the following table] to [**the authorized administrator in the following table**].

TSF data	Ability						Administrator
	change default	query	modify	delete	clear	other operation	
Audit data	-	○	-	-	-	Audit view, statistics	all administrator
User access status	-	○	-	-	-	Force logout	top administrator, user administrator
User information	-	○	○	○	-	Password reset, Unblock	
ID policy	-	○	○	-	-	-	top administrator, policy administrator
Password policy	-	○	○	-	-	-	
End-user policy	-	○	○	-	-	-	
Administrator policy	-	○	○	-	-	allowed ip number	
Audit violation policy	-	○	○		-		
Administrator information	-	○	○	○	-	Password reset, Unblock	
Integrity verification data	-	○	-	-	-	Server/Agent Self-Test	top administrator, system administrator
Business system information	-	○	○	○	-	-	

Group of Business system information	-	○	○	○	-	-	
BATCH information	-	○	○	○	-		
Code-defined	-	○	○	-	-	-	
SQL Query	-	○	○	○	-	-	
End-user's own Password	-	-	○	-	-	-	End-user

[Table 5-11] List of TSF data

Application notes

○ "Manage" to which a refinement operation is applied includes the ability to change default, query, modify, delete, clear, other operation, etc

○ Among the management functions of TSF data specified in above Table, 'Query' function can be performed by all authorized administrators, and 'Modify' and 'Delete' functions can be performed only by the administrator specified in the above table

5.1.4.3 FMT_PWD.1 Management of ID and password (Extended)

Hierarchical to No other components

Dependencies FMT_SMF.1 Specification of Management Functions
FMT_SMR.1 Security roles

FMT_PWD.1.1 The TSF shall restrict the ability to manage the password of [Creating and changing administrator/end-user passwords] to [**the authorized administrator(top administrator, policy administrator)**].

1. [

a) Password combination rule: Two or more combinations of lower case / upper case / numeric / special characters (default: Three combinations of lower case letters, numbers, and special characters)

b) Password length: between 6-99 characters (default: minimum 9 characters, maximum 20 characters)

]

2. [

a) other management such as management of special characters unusable for password: None (32 special characters that can be input by the following keyboard {

'`', '~', '!', '@', '#', '$', '%', '^', '&', '*',
'(', ')', '_', '+', '[', ']', '{', '}', ';', ''',
':', '"', ',', '.', '/', '<', '>', '?', '-', '=',
'_', '+' })

b) Number of repetitions of the same character: Limit more than the number set by the authorized administrator (default 3)

c) Number of consecutive character repetitions: Limit more than the number set by the authorized administrator (default 3)

d) Password change period: The period set by the authorized administrator (1 day ~ 365 days), change password when it is exceeded (default: 60 days)

]

FMT_PWD.1.2 The TSF shall restrict the ability to manage the ID of [Creating and changing administrator/end-user ID] to [the authorized administrator].

1. [

a) ID combination rule: consist of any combination of letters including lower case letters

b) ID length: between 5-99 characters (default: minimum 6 characters, maximum 20 characters)

]

2. [

a) Other management such as management of special characters unusable for ID: Only allowed special characters {minus (-), underline (_)}]

FMT_PWD.1.3 The TSF shall provide the capability for *changing the password when the authorized administrator accesses for the first time.*

5.1.4.4 FMT_SMF.1 Specification of Management Functions

Hierarchical to No other components

Dependencies No dependencies.

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

a) List of security functions specified in FMT_MOF.1

b) Management functions of TSF data specified in FTP_MTD.1

c) Management functions of ID/Password specified in FMT_PWD.1

].

5.1.4.5 FMT_SMR.1 Security roles

Hierarchical to No other components

Dependencies FIA_UID.1 Timing of identification

FMT_SMR.1.1 The TSF shall maintain the roles [

a) top administrator

b) monitoring administrator

c) system administrator

d) policy administrator

e) user administrator

].

FMT_SMR.1.2 The TSF shall be able to associate users and their **roles defined in FMT_SMR.1.1**

5.1.5 Protection of the TSF (FPT)

5.1.5.1 FPT_ITT.1 Basic Internal TSF data transfer protection

Hierarchical to No other components

Dependencies No dependencies.

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

5.1.5.2 FPT_PST.1 Basic protection of stored TSF data (Extended)

Hierarchical to No other components

Dependencies No dependencies.

FPT_PST.1.1 The TSF should protect the [TSF data in the following table] stored in the repository, which is controlled by the TSF, from unauthorized *exposure and modification*.

Storage	TSF data	Protection method

DBMS	User’s(end-user/administrator) password	Access control, Encryption(Hash)
	Among the authentication token components, 'subject (user)'	Access control, Encryption(Symmetric key)
	Server-Agent Shared key(Symmetric key)	Access control, Encryption(Symmetric key)
	Server-Agent Secret key(MAC key)	Access control, Encryption(Symmetric key)
	TOE configuration value (ID / PW generation rule, Authentication failure handling policy)	Access control, Encryption(Symmetric key)
	Audit data	Access control
Filesytem	DBMS account’s password	Encryption(Symmetric key)
	SMTP account’s password	Encryption(Symmetric key)
	TOE’s configuration value (environment configuration parameter)	Encryption(Symmetric key)

[Table 5-12] Protected TSF data

5.1.5.3 FPT_TST.1 TSF testing

Hierarchical to No other components

Dependencies No dependencies.

FPT_TST.1.1 The TSF shall run a suite of self tests *during initial start-up, periodically during normal operation* to demonstrate the correct operation of [*Audit review, Cryptographic support, TSF Generation of secrets, Protection of stored TSF data*].

FPT_TST.1.2 The TSF shall provide **authorized administrators** with the capability to verify the integrity of [*TOE’s environment configuration files*].

FPT_TST.1.3 The TSF shall provide **authorized administrators** with the capability to verify the integrity of [*executable file(Java Archive)*].

5.1.5.1 FPT_TEE.1 Testing of external entities

Hierarchical to No other components

Dependencies No dependencies.

FPT_TEE.1.1 The TSF shall run a suite of tests *at the request of the authorized* ***administrator*** to check the fulfillment of [

the following list of properties of the external entities

- SMTP Server: domain or ip address, service port

].

FPT_TEE.1.2 If the test fails, the TSF shall [the following action(s)

a) SMTP Test failed

- Outputs an error message on security management screen so that authorized administrator can take recovery action.

].

5.1.6 TOE access (FTA)

5.1.6.1 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions

Hierarchical to FTA_MCS.1 Basic limitation on multiple concurrent sessions

Dependencies FIA_UID.1 Timing of identification

FTA_MCS.2.1 The TSF shall restrict the maximum number of concurrent sessions that belong to the same user according to the rules [restriction to one for the maximum number of concurrent sessions for administrator management access session, prohibition of same administrator both concurrent connections of management access session and local access session, the Rules on the maximum number of concurrent sessions { determined as follow }]

{

a) If you log in again with the same account or same privilege, terminate previous connection

b) If the top administrator is accessing the management first, the administrator of the lower privilege is blocked new connection

c) the duplicated logins with the same account and other administrator accounts can be allowed for the monitoring administrator

}

FTA_MCS.2.2 The TSF shall enforce, by default, a limit of [1] sessions per user.

Application notes

○ A session is presented in FMT_MCS.2 is 'administrator access', the number of sessions should be 'the number of administrator accesses.

5.1.6.2 FTA_SSL.5 Management of TSF-initiated sessions (Extended)

Hierarchical to No other components.

Dependencies FIA_UAU.1 Authentication or No dependencies.

FTA_SSL.5.1 The TSF shall *terminate* an interactive session after a [

time interval of administrator/end-user inactivity

a) administrator: fixed value(10 minutes)

b) end-user: Time set by the authorized administrator(1 ~ 60minutes or not used, default value: 10 minutes)

].

5.1.6.3 FTA_TSE.1 TOE session establishment

Hierarchical to No other components.

Dependencies No dependencies

FTA_TSE.1.1 The TSF shall be able to deny **administrator's management access** session establishment based on [connection IP, *whether or not to use, start date, end date*]

Application notes

○ The management access session of administrator shall be allowed only from the terminal with designated IP address for management access.

5.1.7 Trusted path/channels (FTP)

5.1.7.1 FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to No other components.

Dependencies No dependencies

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another 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 *another trusted IT product* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [Send mail with SMTP].

5.2 Security Assurance Requirements

Security assurance requirements of this ST are composed of assurance components in Common Criteria (CC V3.1) Part 3 and the evaluation assurance level is EAL1+(ATE_FUN.1).

The table below summarizes assurance components.

Assurance Class	Assurance Component	
Security Target evaluation	ASE_INT.1	ST introduction
	ASE_CCL.1	Conformance claims
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_ECD.1	Extended components definition
	ASE_REQ.1	Stated security requirements
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic functional specification
Guidance documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-cycle support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_FUN.1	Functional testing
	ATE_IND.1	Independent testing - conformance
Vulnerability assessment	AVA_VAN.1	Vulnerability survey

[Table 5-13] Assurance Component Summary

5.2.1 Security Target evaluation

5.2.1.1 ASE_INT.1 ST introduction

Dependencies No dependencies.

Developer action elements

ASE_INT.1.1D The developer shall provide an ST introduction.

Content and presentation elements

ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C The ST reference shall uniquely identify the ST.

ASE_INT.1.3C The TOE reference shall uniquely identify the TOE.

ASE_INT.1.4C The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C The TOE overview shall identify the TOE type.

ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

Evaluator action elements

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

ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

5.2.1.2 ASE_CCL.1 Conformance claims

Dependencies ASE_INT.1 ST introduction
ASE_ECD.1 Extended components definition
ASE_REQ.1 Stated security requirements

Developer action elements

ASE_CCL.1.1D The developer shall provide a conformance claim.

ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

Content and presentation elements

ASE_CCL.1.1C The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

Evaluator action elements

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

5.2.1.3 ASE_OBJ.1 Security objectives for the operational environment

Dependencies No dependencies.

Developer action elements

ASE_OBJ.1.1D The developer shall provide a statement of security objectives.

Content and presentation elements

ASE_OBJ.1.1C The statement of security objectives shall describe the security objectives for the operational environment.

Evaluator action elements

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

5.2.1.4 ASE_ECD.1 Extended components definition

Dependencies No dependencies.

Developer action elements

ASE_ECD.1.1D The developer shall provide a statement of security requirements.

ASE_ECD.1.2D The developer shall provide an extended components definition.

Content and presentation elements

ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

Evaluator action elements

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

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

5.2.1.5 ASE_REQ.1 Stated security requirements

Dependencies ASE_ECD.1 Extended components definition

Developer action elements

ASE_REQ.1.1D The developer shall provide a statement of security requirements.

ASE_REQ.1.2D The developer shall provide a security requirements rationale.

Content and presentation elements

ASE_REQ.1.1C The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.1.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.1.3C The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.1.4C All operations shall be performed correctly.

ASE_REQ.1.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.1.6C The statement of security requirements shall be internally consistent.

Evaluator action elements

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

5.2.1.6 ASE_TSS.1 TOE summary specification

Dependencies ASE_INT.1 ST introduction
ASE_REQ.1 Stated security requirements
ADV_FSP.1 Basic functional specification

Developer action elements

ASE_TSS.1.1D The developer shall provide a TOE summary specification

Content and presentation elements

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

Evaluator action elements

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

ASE_TSS.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

5.2.2 Development

5.2.2.1 ADV_FSP.1 Basic functional specification

Dependencies No dependencies.

Developer action elements

ADV_FSP.1.1D The developer shall provide a functional specification.

ADV_FSP.1.2D The developer shall provide a tracing from the functional specification to the SFRs.

Content and presentation elements

ADV_FSP.1.1C The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2C The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3C The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements

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

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.3 Guidance documents

5.2.3.1 AGD_OPE.1 Operational user guidance

Dependencies ADV_FSP.1 Basic functional specification

Developer action elements

AGD_OPE.1.1D The developer shall provide operational user guidance.

Content and presentation elements

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements

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

5.2.3.1 AGD_PRE.1 Preprative procedures

Dependencies No dependencies.

Developer action elements

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements

AGD_PRE1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements

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

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.4 Life-cycle support

5.2.4.1 ALC_CMC.1 Labeling of the TOE

Dependencies ALC_CMS.1 TOE CM coverage

Developer action elements

ALC_CMC.1.1D The developer shall provide the TOE and a reference for the TOE.

Content and presentation elements

ALC_CMC.1.1C The TOE shall be labelled with its unique reference.

Evaluator action elements

ALC_CMC.1.1E The evaluator shall confirm that the information provided meet requirements for content and presentation of evidence.

5.2.4.2 ALC_CMS.1 TOE CM coverage

Dependencies	No dependencies.

Developer action elements

ALC_CMS.1.1D	The developer shall provide a configuration list for the TOE.

Content and presentation elements

ALC_CMS.1.1C	The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.
ALC_CMS.1.2C	The configuration list shall uniquely identify the configuration items.

Evaluator action elements

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

5.2.5 Tests

5.2.5.1 ATE_FUN.1 Functional testing

Dependencies	ATE_COV.1 Evidence of coverage

Developer action elements

ATE_FUN.1.1D	The developer shall test the TSF and document the results.
ATE_FUN.1.2D	The developer shall provide test documentation.

Content and presentation elements

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

Evaluator action elements

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

5.2.5.2 ATE_IND.1 Independent testing - conformane

Dependencies ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Developer action elements

ATE_IND.1.1D The developer shall provide the TOE for testing.

Content and presentation elements

ATE_IND.1.1C The TOE shall be suitable for testing.

Evaluator action elements

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

ATE_IND.1.2E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.6 Vulnerability assessment

5.2.6.1 AVA_VAN.1 Vulnerability survey

Dependencies ADV_FSP.1 Basic functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Developer action elements

AVA_VAN.1.1D The developer shall provide the TOE for testing

Content and presentation elements

AVA_VAN.1.1C The TOE shall be suitable for testing.

Evaluator action elements

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

AVA_VAN.1.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

5.3 Security Requirements Rationale

This section sets out the rationale for the dependency on the security functional and warranty requirements to demonstrate that the security requirements described in this ST are appropriate to satisfy the dependency.

5.3.1 Dependency rationale of security functional requirements

The following table shows dependency of security functional requirement.

No.	Security functional requirements	Dependency	Reference No.
1.	FAU_ARP.1	FAU_SAA.1	3
2.	FAU_GEN.1	FPT_STM.1	OE.TRUSTED_TIMESTAMP
3.	FAU_SAA.1	FAU_GEN.1	2
4.	FAU_SAR.1	FAU_GEN.1	2
5.	FAU_SAR.3	FAU_SAR.1	4
6.	FAU_STG.3	FAU_STG.1	OE.TRUSTED_AUDIT_STORAGE
7.	FAU_STG.4	FAU_STG.1	OE.TRUSTED_AUDIT_STORAGE
8.	FCS_CKM.1	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	9 11, 12, 13 10
9.	FCS_CKM.2	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - 8 10
10.	FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	- - 8

11.	FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - 8 10
12.	FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - 8 10
13.	FCS_COP.1(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - 8 10
14.	FCS_COP.1(4)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	- - 8 10
15.	FCS_RBG.1	-	-
16.	FIA_AFL.1(1)	FIA_UAU.1	22
17.	FIA_AFL.1(2)	FIA_UAU.1	22
18.	FIA_IMA.1	-	-
19.	FIA_SOS.1	-	-
20.	FIA_SOS.2	-	-
21.	FIA_SOS.3	FIA_SOS.2	20
22.	FIA_UAU.1	FIA_UID.1	25
23.	FIA_UAU.4	-	-
24.	FIA_UAU.7	FIA_UAU.1	22
25.	FIA_UID.1	-	-
26.	FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	29 30
27.	FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	29 30
28.	FMT_PWD.1	FMT_SMF.1 FMT_SMR.1	29 30
29.	FMT_SMF.1	-	-
30.	FMT_SMR.1	FIA_UID.1	25
31.	FPT_ITT.1	-	-
32.	FPT_PST.1	-	-
33.	FPT_TST.1	-	-
34.	FPT_TEE.1	-	-

35.	FTA_MCS.2	FIA_UID.1	25
36.	FTA_SSL.5	FIA_UAU.1 or No dependencies	22 -
37.	FTA_TSE.1	-	-
38.	FTP_ITC.1	-	-

[Table 5-14] Rationale for the dependency of the security functional requirements

FAU_GEN.1 has a dependency on FPT_STM.1. However, reliable time stamps provided by the security objective OE.TRUSTED_TIMESTAMP for the operational environment of this ST are used, thereby satisfying the dependency.

FAU_STG.3 and FAU_STG.4 have a dependency on FAU_STG.1. However, it is protected from unauthorized deletion or modification in accordance with the security objective OE.TRUSTED_AUDIT_STORAGE for the operational environment of this ST, thereby satisfying the dependency.

5.3.2 Dependency rationale of security assurance requirements

The dependency of EAL1 assurance package provided in the CC is already satisfied, the rationale is omitted

The augmented SAR ATE_FUN.1 has dependency on ATE_COV.1. but, ATE_FUN.1 is augmented to require developer testing in order to check if the developer correctly performed and documented the tests in the test documentation, ATE_COV.1 is not included in this ST since it is not necessarily required to show the correspondence between the tests and the TSFIs.

6. TOE Summary Specification

This chapter specifies the security functionality of the TOE that satisfies the security functional requirements (SFR).

6.1 TOE Security functionality

TOE security functionalities can be roughly divided into "security audit, Cryptographic support, identification and authentication, security management, protection of the TSF protection, TOE access." This section describes how the TOE satisfies the "security functional requirements" specified in the previous section.

6.1.1 Security audit (FAU)

Audit data generation

FAU_GEN.1 The TOE manages all auditable events (defined in FAU_GEN.1) generated during operation through the repository (DBMS) of the TOE operational environment to store audit data.

The audit data generated by the TOE describes date and time of the event, type of event, subject identity, the outcome (success or failure) of the event.

Potential violation analysis and Security alarms

FAU_ARP.1
FAU_SAA.1
The TOE checks the auditable event and notifies the authorized administrator by email if a potential security violation event is detected.

The TOE detects the specified audit event defined in the TOE management tool (audit violation policy management) as a potential security violation by the authorized administrator in the following audit events.

- When the defined number of user and administrator authentication failures is exceeded
- When the product self-test fails (product integrity verification,

cryptographic module self-test)

- When the audit storage threshold is exceeded
- Saturation of audit storage

Audit review

FAU_SAR.1 FAU_SAR.3 The TOE provides the authorized administrator with the ability to distinguish and examine all audit data of the TOE by the audit data type and the selection criteria of type.

The TOE provides search, sorting, and detailed audit data view functions so that it is suitable for the administrator to interpret.

Details of the [Audit review] function of the TOE are specified in **6.1.4.5 A udit view and statistics**.

counteract and prevention of audit data loss

FAU_STG.3 FAU_STG.4 The TOE provides functions to counteract and prevent audit data loss. The counteraction and prevention functions for the audit data loss check the usage amount of the storage at regular intervals to prevent the loss due to the storage capacity shortage. If the use space of the audit trail storage exceeds the specified limit (80% of the database volume capacity), the TOE sends an email to the email account specified by the authorized administrator.

If the audit trail is saturated (the audit trail storage space occupies 95% of the capacity of the database volume), the TOE overwrite the oldest audit record and sends an email to the email address specified by the authorized administrator for countermeasures to recover the audit storage

6.1.2 Cryptographic support (FCS)

Cryptographic key and Random bit generation

FCS_CKM.1 FCS_RBG.1 The TOE generates a secure cryptographic key with the cryptographic complexity of 112 bits or more, using the cryptographic key generation algorithm shown in the following table.

Item	Cryptographic Key generation Algorithm	Detail	Cryptographic Key Size (bits)	List of cryptographic operations
KEK	PBKDF2	HMAC-SHA256 Salt: 128bits Iteration: 1024	128	- private key of the TOE server and TOE agent
Master Key	Hash_DRBG	Hash: SHA-256	256	- Encryption and decryption for Important information of TOE stored data
			160	- Integrity verification of authentication token
Secret Key	Hash_DRBG	Hash: SHA-256	128	- Mutual authentication between TOE components
Shared Key	ECDH	Curve: P-256	256	- Data encryption and decryption between TOE components
			160	- Data integrity verification between TOE components
Elliptic curve key pair	Hash_DRBG	Hash: SHA-256 Curve: P-256	256	- Key exchange between TOE components

[Table 6-1] Cryptographic key generation algorithm

Cryptographic key generation applying password based key derivation functions is used only when generating a key encryption key (KEK). In this case, the message authentication code uses the HMAC-SHA256 algorithm, and the 128-bit random salt value and 1024 iteration counts are applied.

In case of the master key, among the 512 bits obtained by hashing the random value generated by the random bit generator of the validated cryptographic module in the initial setup of TOE by SHA-512, the first 256 bits are used as a symmetric key for the block cipher, and the last 160 bits are used as a MAC key for the integrity verification.

The secret key generates a 128-bit key using the random bit generator of the validated cryptographic module and be used as a MAC key of the HMAC-SHA256 integrity verification algorithm for mutual authentication between the TOE components called Pass-Ni SSO Server and Pass-Ni SSO Agent.

The shared key is used for encryption/decryption of data transmitted and received between the TOE components called Pass-Ni SSO Server and Pass-Ni SSO Agent. The shared key uses the 'ECDH elliptic-curve cryptography (Curve: P-256)' to share mutual cryptographic keys. Among the 512 bits of the shared cryptographic key, the first 256 bits are used as a symmetric key for the block cipher, and the last 160 bits are used as a MAC key for the integrity verification.

When generating a cryptographic key using a random bit, a 'hash-function-based deterministic random bit generator (Hash_DRBG)' specified in 'KS X ISO/IEC 18031' is used, and the hash function generates a random bit using SHA-256.

The cryptographic module used by the TOE uses the validated cryptographic module specified in [Table 1-3] in which the security and implementation conformance are validated by the Korea Cryptographic Module Validation Program (KCMVP).

Cryptographic key distribution

FCS_CKM.2
FCS_COP.1(4)
The TOE components called Pass-Ni SSO Server and Pass-Ni SSO Agent distribute mutual shared keys, using the Elliptic-curve based key establishment mechanism (ECDH, Curve: P-256)' described in 'ISO/IEC 11770-3.'

The ECDH cryptography uses an automatic setting method of an electronic key agreement protocol using the API provided by the validated cryptographic module.

Cryptographic key destruction

FCS_CKM.4 If the TOE is terminated, or when the cryptographic keys loaded onto memory are no longer used, all cryptographic keys loaded onto memory shall be destroyed.

The method of destroying cryptographic keys is the one specified in 'Cryptographic Key Management Guidance (Ministry of Science, ICT, and Future Planning, 2014)', using a method of overwriting all zeros for complete deletion.

Cryptographic operation

FCS_COP.1(1) The TOE performs encryption/decryption with symmetric cryptographic (block cipher) algorithm ARIA-256 when conducting encrypted communication between its components or storing TSF data, and it encrypts/decrypts the private key file of server and agent with SEED algorithm. The operation mode uses the CBC, and the generation of IV (Initialization Vectors) is performed using a method shown in the appendix of "NIST SP 800-38A,' that is, it is generated by using the random bit generator provided by the validated cryptographic module. The list of cryptographic operations performed by the block cipher algorithm is shown in [Table 5-5].

FCS_COP.1(2) The HMAC-SHA256 message authentication code algorithm is used to verify the integrity of the TOE-generated authentication token, and the list of cryptographic operations performed by the HMAC-SHA256 message authentication code is shown in [Table 5-6].

FCS_COP.1(3) The TOE uses the hash function SHA-256 to check the authentication data of the user/administrator and the integrity verification of the TOE, and the list of cryptographic operations performed by the SHA-256 hash function is shown in [Table 5-7].

The cryptographic module used by the TOE for cryptographic operation uses the validated cryptographic module specified in [Table 1-3] in which the security and implementation conformance are validated by the Korea Cryptographic Module Validation Program (KCMVP).

6.1.3 Identification and authentication (FIA)

Identification and authentication

FIA_UID.1
FIA_UAU.1
The TOE identifies all users (authorized administrators and authorized users) accessing it. All users attempting access cannot use any of the functions of the TOE until the user is identified.

The TOE provides identification and authentication using user ID and password. When a user requests the login screen for initial authentication, the TOE creates EC elliptic curve public key pair for confidentiality of the authentication data, generates the nonce value to prevent retransmission attacks, and displays the identification and authentication procedure request screen (login screen). Then, the user receives and verifies the user ID and password input through the identification and authentication procedure request screen (login screen).

The TOE identifies and authenticates the end-user based on ID / PW and issues an authentication token. Thereafter, an authentication token-based authentication is provided so that a end-user can use the service without additional login action when accessing another business system.

FIA_UAU.7 The TOE masks ("● ● ● ● ●") the input password characters and displays them on the screen when inputting confidential information during the authentication procedure. The TOE does not provide a feedback (e.g., an invalid user ID or an incorrect password is entered) on the reason for authentication failure; instead, it generates audit data for the authentication failure.

FIA_UAU.4 The TOE ensures the uniqueness of the session by using the disposable nonce value when authentication and identification are performed by the password-based authentication method, thereby preventing the reuse of the authentication data. The TOE ensures the uniqueness of the authentication token including the nonce value per authentication token and prevents the reuse of the authentication token because the authentication token exceeding the expiration time including the expiration time information cannot be used.

FMT_PWD.1 The TOE enforces the function to change the password when an authorized administrator accesses security management screen for the first time or an end-user accesses at the business system for the first time.

The TOE enforces the function to change the password when an administrator and an end-user perform identification and authentication through the initialized password.

Authentication failure handling

FIA_AFL.1(1)
FIA_AFL.1(2)
FAU_GEN.1

The TOE protects itself from malicious user authentication attempts by providing the user account lock function in [identification and authentication] The TOE does not provide a feedback (e.g., an invalid user ID or an incorrect password is entered) on the reason for authentication failure.

If the number of authentication failures reaches the number of times specified by the authorized administrator (from 3 to 10, default 5 times), it disables the user identification and authentication function of the account. The inactivation of the user identification and authentication function is classified into **temporary block** (from 5 minutes to 60 minutes) set by the authorized administrator or **block** to be disabled until the administrator unlock it. However, The top administrator who is provided at the time of TOE installation disables (temporary block) the identification and authentication function for 5 minutes regardless of the administrator's action when the number of authentication failures is reached. If the user tries to authenticate through the blocked account, the TOE processes the authentication failure and generates audit data.

The authorized administrator can inquire or unlock the blocked user account through the administrator management and user management functions. A user whose account is unblocked can use the functions provided by the TOE normally when the [identification and authentication] is successfully completed.

Verificatin of secrets

FIA_SOS.1 The TOE performs an automated inspection that checks whether it meets permission criteria for secrets set by an authorized administrator when an administrator and an end-user create or change a password or when an authorized administrator changes the default password provided at initial connection

FMT_PWD.1 The TOE provides an authorized administrator with the function of setting the acceptance criteria of secrets such as the allowable characters for user IDs/combination rule/length of and allowable characters for password/combination rule/length of administrator and end-user.

Generation and Destruction of authentication token

FIA_SOS.2 Among the TOE components, Pass-Ni SSO Server verifies (identifies and authenticates) the authentication information entered from the user's login screen and generates and issues an authentication token. The authentication token issued is encrypted and passed to the Pass-Ni SSO Agent.

The structure of the authentication token conforms to the 'RFC7519 JSON Web Token (JWT)' standard. The authentication token has three elements: a header, a payload, and a signature, which are separated by a dot(.), and each has a form of 'xxxxxx.yyyyyy.zzzzzz' encoded by base64Url. The header and payload parts of the authentication token are guaranteed to be integrity, and the payload part is encrypted with the validated cryptographic module to ensure confidentiality.

FIA_SOS.3 The TOE terminates the session and deletes the authentication token loaded on the memory when it receives a logout request by the user or the user is inactive for a certain time interval (inactivity period set by the authorized administrator. Default: 10 minutes) after the login.

The authentication token is destroyed by initializing the variable assigning the authentication token to be null.

TOE Internal mutual authentication

FIA_IMA.1
FCS_CKM.2
FAU_GEN.1

The TOE sets a mutual cryptographic key by the 'ECDH (Elliptic Curve Diffie-Hellman) elliptic-curve cryptography (Curve: P-256)' to share the mutual cryptographic key between the components called Pass-Ni SSO Server and Pass-Ni SSO Agent. The mutual authentication between the components is performed through **in-house implementation Handshake Authentication Protocol** using HMAC-SHA256 integrity verification algorithm.

The integrity verification algorithm used in the internally implemented authentication protocol mechanism employs the HMAC-SHA256 algorithm, and the 128-bit secret key using the random bit generator of the validated cryptographic module in the preparation step is respectively loaded when the TOE_Server and TOE_Agent are installed.

The mechanism of the internally implemented authentication protocol for mutual authentication between TOE components is as follows.

1) TOE_Agent generates a random bit and sends it to TOE_Server.
2) The TOE_Server generates the integrity verification code by using the bit of the TOE_Agent as the secret key, generates the integrity verification code value and a random bit, and transmits it to the TOE_Agent.
3) The TOE_Agent verifies the integrity verification code (MAC) with the secret key, generates the integrity verification code in the bit of TOE_Server, and transmits it to the TOE_Server.
4) TOE_Server verifies the integrity verification code with the secret key.

The TOE generates audit data on success/failure events of mutual authentication between the components.

6.1.4 Security management (FMT)

After the TOE has been properly installed, the authorized administrator can access the TOE security management interface (GUI) through the web browser (for example, Internet Explorer 11) from the management PC of the IP allowed explicit access. The TOE allows access to the security management interface (HTTPS) only when the user attempting connection has successfully completed the identification and authentication procedure enforced by the TOE.

Security roles

FMT_SMR.1 The roles of an authorized administrator of the TOE are divided into five types: top administrator, monitoring administrator, system administrator, policy administrator, and user administrator.

a) top administrator: an authorized administrator who has full authority and has been granted the authority of all the security management functions provided by the TOE.
b) monitoring administrator: an administrator who has been granted the authority of the inquiry functions among the security management functions provided by the TOE.
c) system administrator: an administrator who have been granted management authority for business system management, system checking, BATCH management, SQL management, and code management functions.
d) policy administrator: an administrator who have been granted management privileges for ID and password policy management, user and administrator policy management, audit violation policy management, and administrator management functions.
e) user administrator: an administrator who has been granted the authority of the user management function such as inquiry function, user information inquiry and password reset.

System administrator, policy administrator, and user administrator are granted the authority to inquiry all the security management functions provided by the TOE

Since all administrator IDs except for the top administrator are not provided as default in the TOE and only their roles are defined, the top administrator shall register new administrators when necessary and grant them the necessary authorities.

FAU_GEN.1 The TOE generates audit data about the management actions when the authorized administrator performs the security management function.

6.1.4.1 Link System management

The TOE performs [link system management] function for administrating the link system (business system) in which the SSO agent is installed.

Link System Management

FMT_MOF.1
FMT_SMF.1
FMT_MTD.

The TOE provides [link system management] function to register, modify, or delete the link system to the authorized administrator. In order for a general user to log in to the business system using the Single Sign On service, the authorized administrator shall register the link system through this function.

The authorized administrator manages the ID of link system, system name, character set, link system identification code, enabled/disabled, domain, IP, and link system public key information through the security management interface. The TOE provides Single Sign On service to end-users using the link system information registered by the authorized administrator.

6.1.4.2 Policy management

The TOE performs [policy management of user ID and password], [policy management of user and administrator], and [policy management of audit violation] functions to manage the operational policy of itself.

Policy management of user ID and password

FMT_MOF.1
FMT_SMF.1
FMT_MTD.1
FMT_PWD.1

The TOE provides the authorized administrator with the [policy management of user ID and password policy] function which manages the allowable characters, combination rule, and length for each ID and password.

The TOE provides the function of managing the following user ID policy to the authorized administrator and enforces the permissible characters, the combination rule, and the minimum/maximum length verification criteria when creating the administrator and user ID.

a) Character count: minimum count (default: 6, input range: 5 to maximum count), maximum count (default: 20, input range: minimum count to 99)
b) Upper case inclusion: enabled/disabled (default: disabled)
c) Lower case inclusion: enabled/disabled (default: enabled)
d) Number inclusion: enabled/disabled (default: disabled)

The TOE provides the function of managing the following password policy to the authorized administrator and enforces each verification criterion of allowable character, combination rule, minimum/maximum length, and change interval for password when creating or changing administrator and user password.

a) Character count: minimum count (default: 9, input range: 6 to maximum count), maximum count (default: 20, input range: minimum count to 99)
b) Same character: enabled/disabled, repeating count (default: 3, input range: 2-9)
c) Consecutive characters: enabled/disabled, consecutive number of times (default: 3, input range: 2-9)
d) Upper case inclusion: enabled/disabled (default: disabled)
e) Lower case inclusion: enabled/disabled (default: enabled)
f) Number inclusion: enabled/disabled (default: enabled)
g) Special character inclusion: enabled/disabled (default: enabled)

Policy management of user and administrator

FMT_MOF.1
FMT_SMF.1
FMT_MTD.1
FIA_AFL.1
FTA_TSE.1

The TOE provides the authorized administrator with the [policy management of user and administrator] function that manages the number of failed password entry (number of authentication failures) and the password change interval for each user and administrator.

The TOE provides the function of managing the following user policy to the authorized administrator and enforces each verification criterion of the user policy such as the number of failed password entry and password change interval during user identification and authentication.

a) Failed password entry: enabled/disabled, failure count (default: 5, input range: 3-10), block/temporary block, duration of blocks (default: 5 min., input range: 5-60 min.)

b) Password change interval: enabled/disabled, change interval (default: 60 days, input range: 1-365 days)

c) Automatic logout: enabled/disabled, time of inactivity (default: 10 min., input range: 1-60 min.)

d) Duplicate login restrictions: enabled/disabled

e) Authentication token valid time: valid time (default: 300 min., input range: 30-1,000 min.)

Policy management of audit violation

FMT_MOF.1
FMT_SMF.1
FMT_MTD.1
FAU_SAA.1

The TOE inspects the auditable events, detects potential security violation events, and notifies the authorized administrator by email. The TOE provides the function of [policy management of audit violation] that can manage the email notification about each security violation event.

The following are events that can be managed by the TOE management function to notify the email notification of the security violation event.

- Administrator authentication failure times exceeded (default: enabled)
- End-user authentication failure times exceeded (default: enabled)
- Internal server test failed (default: enabled)
- Internal Agent test failed (default: enabled)
- Audit storage threshold exceeded (default: enabled)
- Saturation of audit storage (default: enabled)

6.1.4.3 End-user management

The TOE performs [user management] and [administrator management] functions for user management of itself.

End-user management

FMT_MOF.1
FMT_SMF.1
FMT_MTD.1

TOE provides [end-user management] function to manage end-users who use Single Sign On service. The authorized administrator can view the currently connected users through the security management interface and can enforce the users to log out by selection.

The TOE provides the function to view the registered users to the authorized administrator or to register, modify, and delete end-users. The authorized administrator can manage the name, telephone number, email information, start/end date, and active/inactive status of end-users. In addition, it provides functions to initialize the password of the user who lost the password or to unlock the inactivation of authentication and identification function caused by the exceeded number of authentication failures (default: 5 times).

The TOE provides the ability to change its password to the end-user.

Administrator management

FMT_MOF.1 The TOE provides [administrator management] function to
FMT_SMF.1 view/register/modify/delete administrators that can access through the
FMT_MTD.1 security management interface. The TOE provides the top administrator as a default, authorized administrators can view the registered administrators, register new administrators, or delete the existing administrators.

The authorized administrator can manage the name, phone number, e-mail information, start/end date, notification email reception, and active/inactive status of administrators. If the start/end date of use is specified, the security management interface can be accessed only for a specified period. When inactivated, security management access (identification and authentication) of the administrator is restricted.

FMT_SMR.1 The authorized administrator can select the administrator role defined in the TOE at administrator registration/modification. There are five types of authorized administrators of the TOE: top administrator, monitoring administrator, system administrator, policy administrator, and user administrators.

FTA_TSE.1 The TOE provides a function to register/modify/delete access permitted IPs for each administrator as the number of access permitted IPs (default: 2) set in the [policy management of administrator] function through the [administrator management] function. The administrator can only access the terminal registered as the access permitted IP when requesting identification and authentication.

6.1.4.4 System management

The TOE performs [system check], [batch management], [SQL management], and [Code Management] functions for managing the TOE system.

System check

FMT_MOF.1
FMT_SMF.1
FPT_TST.1
FPT_TEE.1
FAU_GEN.1

The TOE performs self-tests during initial start-up or periodically to demonstrate the correct operation of the TSF and, in addition, provides the authorized administrator with a [system check] function to perform self-tests and tests of external entities, if necessary.

Self-tests consist of server self-tests and agent self-tests. The server self-test and the agent self-test perform self-tests such as integrity verification and cryptographic module self-test of the SSO server and agent, respectively, when an authorized administrator requests it and generate audit data on the result of the execution.

The SMTP server test is performed in the test of external entities. The SMTP test checks the SMTP server's domain or IP address, and access to the service port when an authorized administrator requests it and generates audit data on the result of the execution.

Batch management

FMT_SMF.1
FMT_MTD.1

The TOE provides [batch management] function to manage functions executed periodically during TOE operation. The authorized administrator can set the execution cycle and active status of the following functions through the security management interface.

a) garbage collection: clear memory by the garbage collection
b) Database capacity check: capacity check on audit trail storage
c) Server self-test: integrity verification, cryptographic module self-test
d) Agent self-test: integrity verification, cryptographic module self-test

SQL management

FMT_SMF.1
FMT_MTD.1

The TOE provides [SQL management] function for managing the SQL syntax used during TOE operation. The authorized administrator can manage the SQL query statement, SQL data types, and active status through the security management interface. There are four types of SQL, 'R, I, U, D', which indicate SELECT, INSERT, UPDATE, and DELETE, respectively.

Code management

FMT_SMF.1
FMT_MTD.1

The TOE provides the [code management] function to manage the codes used in the TOE operation. The authorized administrator can manage the audit types, error messages(result code), maile types, characterset types, development language types through the security management interface.

6.1.4.5 Audit view and statistics

The TOE performs [system audit review], [audit review of link system], [audit review of administrator access and operation], [audit review of user access and operation], and [statistics review].

System audit review

FAU_SAR.1
FMT_MOF.1
FMT_SMF.1

The TOE provides the authorized administrator with [system audit review] function to check the operational status of the TOE. The types of system audit list that the authorized administrator can view from the security management interface are as follows.

a) Start of a server
b) Termination of a server
c) Issue of an authentication token
d) Destruction of an authentication token
e) Server module tests
f) Server integrity verification
g) (Server) cryptographic module self-test

FAU_SAR.3

The TOE views the audit data based on [Table 5-3] the selection criteria by the types of audit data and sequences them in descending order based on the occurrence time. The authorized administrator can sequence them from the viewed audit list in descending order or ascending order based on serial numbers, types, result codes, and occurrence time.

Audit review of link system

FAU_GEN.1
FAU_SAR.1
FMT_MOF.1
FMT_SMF.1

The TOE generates audit data on the result of interoperating with the SSO agent installed in the link system. The TOE provides the authorized administrator with the [audit review of link system] function that can view the details of the interoperating of the link system. The types of link system audit lists that an authorized administrator can view from the security management interface are as follows.

a) Login request
b) Logout request
c) Request for user information
d) Request for security policy information
e) Request for duplication login check
f) Request a key exchange
g) Agent module test
h) Agent integrity verification
i) (agent) cryptographic module self-test

FAU_SAR.3

The TOE views the audit data based on [Table 5-3] the selection criteria by the types of audit data and sequences them in descending order based on the occurrence time. The authorized administrator can sequence them from the viewed audit list in descending order or ascending order based on serial numbers, types, names of link system, and occurrence time.

Audit review of administrator access and operation

FAU_GEN.1
FAU_SAR.1
FMT_MOF.1
FMT_SMF.1

The TOE generates audit data on the execution results when the authorized administrator accesses the security management interface or performs security management functions. The TOE provides the [audit review of administrator access and operation] function which can view the access and the operation history of the authorized administrator.

FAU_SAR.3

The TOE views the audit data based on [Table 5-3] the selection criteria by the types of audit data and sequences them in descending order based on the occurrence time. The authorized administrator can sequence them from the viewed audit list in descending order or ascending order based on serial numbers, types, user IDs, names, and occurrence time.

Audit review of user access and operation

FAU_GEN.1
FAU_SAR.1
FAU_SAR.3
FMT_MOF.1
FMT_SMF.1

The TOE generates audit data on execution results on the access (identification and authentication) of end-users, password change, and logout. The TOE provides the [audit review of user access and operation] function that can view the access and the operation history of the end-users.

FAU_SAR.3

The TOE views the audit data based on [Table 5-3] the selection criteria by the types of audit data and sequences them in descending order based on the occurrence time. The authorized administrator can sequence them from the viewed audit list in descending order or ascending order based on serial numbers, types, user IDs, names, and occurrence time.

Statistics review

FMT_SMF.1

The TOE provides yearly and monthly user access statistics based on the user access audit data generated during its operation.

6.1.5 Protection of the TSF (FPT)

TSF data transfer protection

FPT_ITT.1
FCS_CKM.2
FCS_COP.1(1)

The TOE sets a mutual cryptographic key by the ‘ECDH (Elliptic Curve Diffie-Hellman) elliptic-curve cryptography (Curve: P-256)’ to share the mutual cryptographic key between the components called Pass-Ni Server and Pass-Ni Agent. The mutual authentication between the components is performed through in-house implementation Handshake Authentication Protocol using HMAC-SHA256 integrity verification algorithm. The sharing of the cryptographic key between the TOE components is performed when the Pass-Ni Agent is startup.

The transmission data to be transmitted between the TOE components is sent and received by encrypting with the approved algorithm (default: ARIA256) using the shared cryptographic key in the above key sharing process.

Protection of stored TSF data

FPT_PST.1 The TOE protects '[Table 5-12] Protected TSF data' against unauthorized exposure and modification by using access control, encryption (hash, symmetric key). The TSF data stored in the DBMS is protected against unauthorized access by using the protection function of the DBMS which provides the identification and authentication and access control functions, and it is also protected through encryption by the approved cryptographic algorithm of validated cryptographic module. The TSF data of the configuration file stored in the file system are protected through encryption by the approved cryptographic algorithm of validated cryptographic module.

The passwords of end-users and administrators are encrypted with the approved hash function (default: SHA256) of the validated cryptographic module including a random 32-character salt string. The TOE uses the message authentication code (HMAC-SHA256) to ensure the integrity of the authentication token and encrypts the authentication token stored in the server repository with the approved cryptographic algorithm (default: ARIA-256). The Secret key used in the server-agent mutual authentication and the encryption key shared between the server and the agent in case of agent activation are stored in the server storage and encrypted by the approved cryptographic algorithm (default ARIA-256). Besides, DB cryptographic key, key encryption key (KEK), DBMS account password, SMTP account password, and TOE configuration value (configuration parameters) stored in the file system can be stored after being encrypted with the approved cryptographic algorithm (default: ARIA-256).

The key encryption key is a cryptographic key generated by the PBKDF2, a key derivation function, from the password entered by the installer at the time of initial installation of the TOE. It is encrypted with the approved cryptographic (default: ARIA-256) and stored in the file system. The PBKDF2 uses the HMAC-SHA256 algorithm, and the random 128-bits salt value and the 1024-times iteration is applied. The secret key employed when encrypting the key encryption key is used by collecting the information of operational environment after installing TOE.

The cryptographic key is loaded into memory by XOR operation with an arbitrary salt value to protect against attacks from memory dumps. The salt value is an arbitrary bit number generated by an approved random bit generator when the server is operating. It is loaded into memory in the form of binary data (byte array) to protect against attacks from string dumps. The cryptographic key is restored to the plaintext with the salt value and XOR operation when it is used in the encryption/decryption operation, and the restored cryptographic key is discarded by the cryptographic key destruction method specified in FCS_CKM.4 after the encryption/decryption operation.

Self-test

FPT_TST.1 In order to demonstrate the correct operation of the security functions, the
FAU_ARP.1 TOE performs server self-tests periodically (default: 24 hours) during normal
FAU_GEN.1 operation at the initial TOE startup. The test interval can be set by an
FAU_SAA.1 authorized administrator through the TOE security management interface.
The self-test performs the integrity verification and the cryptographic module self-test for SSO server and agent, respectively

Component	Type	Target Files
SSO Server	Executable Code	passni-sso-server-4.0.{distribution version}.jar {webhome}/WEB-INF/app/adm/*.* {webhome}/WEB-INF/app/usr/*.*
	Configuration File	crypto-config-properties.xml dbms-config-properties.xml mail-config-properties.xml sso-config-properties.xml
	Cryptographic Library	{crypto-installdir}/*.dll {crypto-installdir}/*.so
SSO Agent for JAVA	Executable Code	passni-sso-agent-4.0.{distribution version}.jar
	Configuration File	pni-config-properties.xml
	Cryptographic Library	{crypto-installdir}/*.dll {crypto-installdir}/*.so

SSO Agent for ASP.NET	Executable Code	Ubintis.PassNi.SSO.dll
	Configuration File	pni-config.xml
	Cryptographic Library	{crypto-installdir}/*.dll {crypto-installdir}/*.so
SSO Agent for PHP	Executable Code	{webhome}/passni/sso/*.*
	Configuration File	pni-config.xml
	Cryptographic Library	{crypto-installdir}/*.dll {crypto-installdir}/*.so

[Table 6-2] Files to be verified for integrity

The cryptographic module self-test verifies whether the validated cryptographic module, whose security and implementation conformance have been validated through the Korea Cryptographic Module Validation Program (KCMVP), is operating normally. The test is conducted by calling the self-test API interface of the validated cryptographic module. If an error is found, the TOE notifies it email set by an authorized administrator and generates audit data.

The authorized administrator can perform the server self-test through the TOE security management interface. If the test fails, the TOE notifies it by email specified by an authorized administrator and generates the audit data.

An authorized administrator can take countermeasures such as ignoring or rebooting the server. It generates audit data for all these events.

Testing of external entities

FPT_TEE.1
FAU_GEN.1

In order to demonstrate the correct operation of the security alert (notification email transmission) function, the TOE performs a test of external entities to check whether the domain address and the service port of SMTP server linked with the TOE are normally connected when an authorized administrator requests it. If the SMTP test fails, audit data is generated, and an error message is displayed on the security management screen so that the authorized administrator can take a recovery measure.

6.1.6 TOE access (FTA)

Administrator session management

The TOE provides a web-based interface for authorized administrators to connect using a web browser installed on the administrator PC when TOE is normally delivered/installed. The TOE allows access to the management access interface (HTTPS) only when the user attempting to connect has successfully completed the administrator identification and authentication process

FTA_TSE.1 The TOE allows or blocks access to the management access interface based on connection IP, active status, and the period of use (start/end date) of an authorized administrator. The TOE rejects the management access session when the connection IP is not registered by an administrator accessing the management access interface, the usage period is not reached or exceeded, or it remains inactive.

FTA_SSL.5 The TOE terminates the session that interacts with an authorized administrator when the authorized administrator has successfully logged in to the management interface (Web UI) of the TOE and exceeded the time interval of user inactivity. The allowed time interval of user inactivity has the fixed value of 10 minutes and cannot be changed. If the authorized administrator tries to use the security management interface again at the end of the session, the TOE moves to the management access initial screen (login screen) and the administrator shall perform administrator re-authentication (user identification and authentication).

FTA_MCS.2 The TOE limits the maximum number of multiple concurrent access sessions to 1 for administrator management access sessions. Accordingly, the same user cannot connect to two or more management access interfaces at the same time, and the TOE terminates the previous session when the same account or the administrators with the same authority simultaneously connect to the management access interface. If the top administrator connected to the management access first, it blocks new access of the administrators (system administrator, administrator, user administrator) having lower authority. However, the duplicated login can be allowed for the monitoring administrator.

FAU_GEN.1 The TOE generates audit data on the results of these events, that is, the execution results of [administrator management access session].

User session management

FTA_SSL.5
FAU_GEN.1 The TOE provides the authorized administrator with the capability to set the inactivity time of an end-user (1-60 min. or inactive, default: 10 min.). The TOE terminates the session when the end-user is inactive during the time interval of user activity set by the authorized administrator after login and generates audit data on the execution result. The TOE shall prevent the end-user from accessing the personalization function screen such as password change at the end of the session, and the end-user shall perform the re-authentication (identification and authentication).

6.1.7 Trusted path/channels (FTP)

Inter-TST trusted channel

FTP_ITC.1 The TOE sends and receives data using the trusted channel employed the TLS 1.2 (RFC 5246) protocol to protect the channel data from modification or disclosure of a communication channel between the TOE and the SMTP server.

The TOE performs TLS communication with the SMTP server using JSSE (Java Secure Socket Extension).