KAYTUS Server Baseboard Management Controller

Security Target

Evaluation Assurance Level (EAL): EAL2 + ALC_FLR.2

Version: 0.9
17 July 2024

Prepared by:

KAYTUS SYSTEMS PTE. LTD.
22 SIN MING LANE,
#06-76, MIDVIEW CITY,
SINGAPORE (573969)

DOCUMENT HISTORY

Rev.	Issue Date	Description	Author
0.1	13 March 2023	Initial draft	Alisha Zhao
0.2	06 June 2023	Revised version	Alisha Zhao
0.3	22 June 2023	Included crypto information	Alisha Zhao
0.4	12 September 2023	Minor corrections	Alisha Zhao
0.5	14 November 2023	Corrections based in the ORs	Alisha Zhao
0.6	22 December 2023	Final corrections	Alisha Zhao
0.7	22 April 2024	Corrections CB	Alisha Zhao
0.8	25 June 2024	Corrections CB	Alisha Zhao
0.9	17 July 2024	Corrections CB	Alisha Zhao

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 SECURITY TARGET INTRODUCTION

This Security Target (ST) defines the scope of the evaluation in terms of the assumptions made, the intended environment for the Target of Evaluation (TOE), the Information Technology (IT) security functional and assurance requirements to be met, and the level of confidence (evaluation assurance level) to which it is asserted that the TOE satisfies its IT security requirements. This document forms the baseline for the Common Criteria (CC) evaluation.

1.1 DOCUMENT ORGANIZATION

Section 1, ST Introduction, provides the Security Target reference, the Target of Evaluation reference, the TOE overview and the TOE description.

Section 2, Conformance Claims, describes how the ST conforms to the Common Criteria and Packages. The ST does not conform to a Protection Profile.

Section 3, Security Problem Definition, describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions applicable to this analysis.

Section 4, Security Objectives, defines the set of security objectives to be satisfied by the TOE and by the TOE operating environment in response to the problem defined by the security problem definition.

Section 5, Extended Components Definition, defines the extended components added to the CC part 2 in terms of security functional requirements.

Section 6, Security Requirements, specifies the security functional and assurance requirements that must be satisfied by the TOE and the IT environment.

Section 7, TOE Summary Specification, describes the security functions that are included in the TOE to enable it to meet the IT security functional requirements.

Section 8, Terminology and Acronyms, defines the acronyms and terminology used in this ST.

1.2 SECURITY TARGET REFERENCE

ST Title: KAYTUS Server Baseboard Management Controller Security Target

ST Version: 0.9

ST Date: 17 July 2024

1.3 TOE REFERENCE

TOE Name: KAYTUS Server Baseboard Management Controller

TOE Version: 1.49.07

TOE Developer: KAYTUS SYSTEMS PTE. LTD

1.4 TOE OVERVIEW

The **KAYTUS Server Baseboard Management Controller** (also referred to as TOE) is an out-of-band management firmware running in a System-on-Chip microprocessor located in an KAYTUS V2 Server (hereinafter referred to as "host") that provides remote management capabilities, including hardware asset management, health status monitoring, fault analysis and remote control..

The TOE runs on an integrated System-on-Chip microprocessor for the remote monitoring/control system. The microprocessor co-exists on the system board with the managed server. The TOE functions independently of the server's state of operation, and the state of the server itself is transmitted to microprocessor through the internal hardware interface.

The TOE can be used in the following situations:

The operation and health status of key hardware components in the host can be monitored through the TOE GUI in real time;
Hosts can be remotely managed through TOE, for startup, shutdown, firmware deployment, and update operations;
Automated large scale multi-server maintenance can be achieved utilizing Redfish API capability offered from the TOE.

Remote administration communication is protected using cryptography. The TOE offers WebUI (Web browser) and Redfish API management interfaces protected by HTTPS/TLS v1.2 and v1.3, and command line console access (SMASH CLP CLI) protected by SSHv2. The remote management functionality provided by the TOE is access controlled and administrator actions are audited.

As summary, the TOE provides the next major security features:

Auditing: the TOE generates audit data during its operation using reliable timestamps, and provides audit review capabilities. The audit data is securely stored.
Cryptographic operations: the TOE enforces several cryptographic algorithms for protecting the external communications and for integrity checks using digital signature algorithms.
Access control policy: the TOE implements access control policy for ensuring that only allowed roles perform selected actions.
Identification and authentication: the TOE includes identification and

authentication capabilities for ensuring that any user is accountable for their actions.
Security management: the TOE provides several management functions accessible for different enabled user roles. The TOE also provides capabilities for user roles management.
Protection of the TSF: the TOE has the ability to protect itself by implementing a set of actions after failure, including recovery after booting and firmware update.
Session management: the TOE is able to manage sessions established with third parties, including session establishment and session termination.
Trusted path: the TOE establishes a trusted path with third parties that ensures confidentiality, integrity and authentication.

1.4.1 TOE Type

The TOE is an out-of-band management firmware that acts as the remote management system for the KAYTUS V2 Server family.

1.4.2 Non-TOE hardware/software/firmware

The Figure 1 - TOE Environment shows the non-TOE HW, SW and FW items:

The board where the TOE is running is an advanced RISC1 machine (ARM) and uses the AST2600 server management processor.

The hardware (KAYTUS V2 Server) where the microprocessor coexists is the item placed at the left of the figure below.

An administrator terminal which is a Windows 10 x64 general purpose computer is required to manage the TOE.

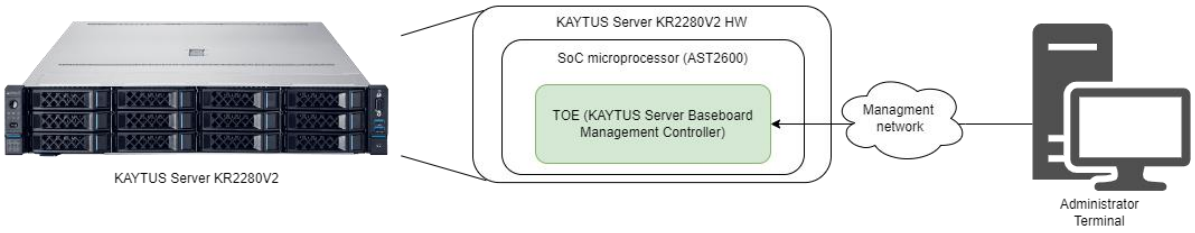

Figure 1 - TOE Environment

The administrator terminal is used to access the TOE's out-of-band management module through the management network using a web browser, Redfish API client, or SSH client. The following third-party software is required when interfacing with the TOE

Java Runtime Environment: OpenJDK 1.8+

SSH Client: PuTTY Version 0.76+

Redfish API Client: Postman Version 7+

Web browsers: Google Chrome 58+

1.5 TOE DESCRIPTION

1.5.1 Physical Scope

1.5.1.1 TOE Delivery

The chassis contains the Host hardware, the microprocessor and the firmware (the TOE) that is shipped directly to customers. The TOE is pre-installed in KAYTUS premises and delivered to customers after the installation. So, the TOE is already installed when the customers receive the chassis.

Alternatively, the evaluated version of the firmware and guidance documentation may be downloaded from the support site:

https://www.kaytus.com/

From there select Support -> Documentation for the documentation or Support -> Drivers for the firmware, then select the desired KAYTUS server model.

TOE	Version	Format	SHA256 digest
KAYTUS Server Baseboard Management Controller (ISBMC_EagleStream_1.49.07_Standard_20240416.hpm)	1.49.07	HPM	3a4959015b1495b67cba6f2c873a9952a7b3ef51ffb8562dd0aa2bde98b15556

1.5.1.2 TOE Guidance

The TOE includes the following guidance documentation, which can be downloaded by the user from the KAYTUS website:

Document	Version	Format	SHA256 digest
Kaytus Server BMC User Manual for KAYTUS V2 Series Servers_Powered by Intel Processors	1.4	PDF	5d4d59101807ec2809ece7f919d35f12b3d209d45fbca4be2e6b2124fd63d197
Kaytus Server BMC Configuration Manual for KAYTUS V2 Series Servers_Powered by Intel Processors	1.2	PDF	f20391d5a67d96a4c75f7ffcd530d493783278084d6a764bcf9b9fcb98074ad5
Kaytus Server BMC Update Manual for KAYTUS V2 Series	1.2	PDF	685ee0b4e1ab7f043f396bebeb40a776a54ae0e5949

Servers_Powered by Intel Processors			093959ddf043bd966066d
Redfish User Manual for KAYTUS V2 Series Servers_Powered by Intel Processors	1.2	PDF	9d989983ce90dc9de7e5e48059ccabdd7fa7730d4d394f15ac1109dc3d6242c2
KAYTUS Server Baseboard Management Controller Common Criteria Guidance Supplement	0.9	PDF	0ad438213e7edd22f9698a47bb9354baf80fd81a83f15d45fb9120c6916231a4

1.5.1.3 TOE evaluated configuration

The evaluated configuration is shown in **Figure 2 – Evaluated Configuration** and consists of the TOE, the microprocessor and the Host hardware.

The microprocessor with the TOE is preinstalled within the Host's chassis. The microprocessor hardware is an advanced RISC1 machine (ARM) and uses the AST2600 server management processor. The microprocessor is managed through external network interfaces, and it communicates with the host with internal circuit board connections.

The administrator terminal is set as indicated in section **1.4.2 Non-TOE hardware/ software/firmware**.

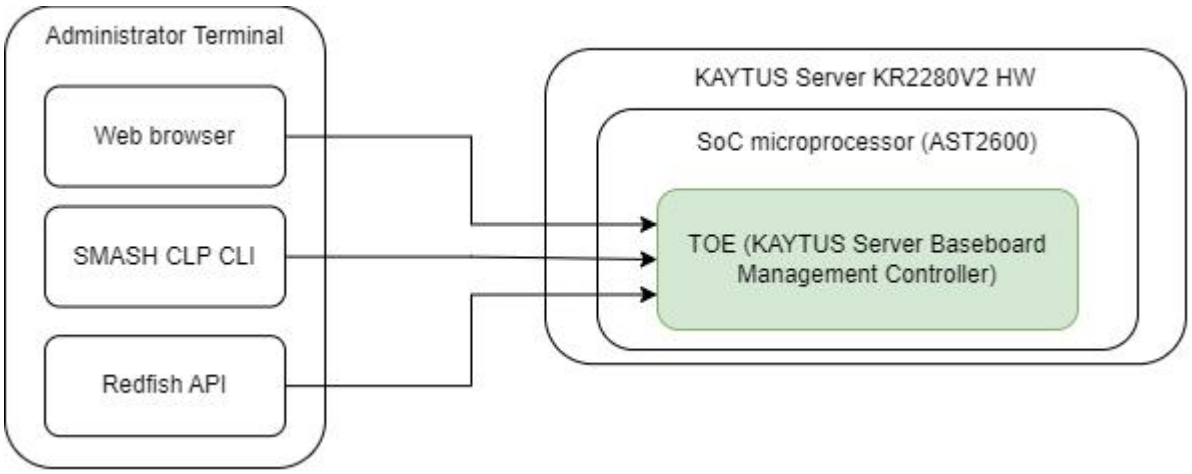

Figure 2 – Evaluated Configuration

1.5.2 Logical Scope

The TOE supports a large range of security functionalities to facilitate various IT systems integration, yet not all these functionalities are in the scope of the ST. As some of them are obsolete and may not be secure enough, or rarely used in most scenarios, they are preserved for legacy purposes on old IT systems which still did not migrate to state-to-art technologies. Nonetheless, they shall not be enabled in this certified configuration.

The next table shows the list of secure functionalities/protocols and the ones that should be enabled in the evaluated configuration of the TOE.

Security Functional Specification		Enabled in TOE
TOE management	WebUI (Web User Interface)	Y
	SMASH CLP CLI (Command Line Interface)	Y
	Redfish API	Y
	IPMI	N
	SNMP	N
User authentication mechanism	Password	Y
	LDAP/AD server	N
	Two Factor Authentication (by certificates)	N
Security protocols	TLS1.3/1.2	Y
	TLS1.1/1.0	N
	SSH V2	Y

Table 1 – Secure functionalities and protocols

The logical boundary of the TOE includes all interfaces and functions within the physical boundary. The logical boundary of the TOE may be broken down by the security function classes described in Section 6. Table 2 summarizes the logical scope of the TOE.

Functional Classes	Description
Security Audit	Audit entries are generated for security related events. The audit logs are protected from unauthorized modification and deletion and may be reviewed by authorized administrators. Time stamp information is provided to support auditing.
Cryptographic Support	Cryptographic functionality is provided to allow the communications links between the TOE and its remote administrators to be protected. Digital signature algorithms are also implemented for firmware integrity checks.
User Data	The TOE provides a role-based access control capability to ensure that only

Functional Classes	Description
Protection	authorized administrators are able to administer the TOE.
Identification and Authentication	Users must be identified and authenticated prior to gaining access to the TOE.
Security Management	The TOE provides management capabilities via a Web-Based GUI, accessed via HTTPS (TLSv1.2 and TLSv1.3). Management functions allow the administrators to configure system and network settings, configure users and roles, and manage the host. The TOE can also be managed over SSHv2 using the SMASH CLP CLI or the Redfish API using HTTPS. (TLSv1.2 and v1.3).
Protection of the TSF	The TOE provides time stamps for the audit records and its own use. The TOE implements actions after failures in the firmware integrity check.
TOA Access	TOE can lock an interactive session and allow user-initiated termination of the user's own interactive session. A TOE administrator may configure to deny session establishment.
Trusted Path/Channel	The communications links between the TOE and its remote administrators are protected using HTTPS (TLSv1.2 and v1.3) and SSHv2.

Table 2 – Logical Scope of the TOE

1.5.3 Functionality Excluded from the Evaluated Configuration

The following features are excluded from this evaluation:

- IPMI
- SNMP
- LDAP/AD server
- Two factor authentication
- NTP
- VNC

2 CONFORMANCE CLAIMS

2.1 COMMON CRITERIA CONFORMANCE CLAIM

This Security Target claims to be conformant to Version 3.1 of Common Criteria for Information Technology Security Evaluation according to:

> Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2017-04-001, Version 3.1, Revision 5, April 2017
>
> Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2017-04-002, Version 3.1, Revision 5, April 2017
>
> Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components CCMB-2017-04-003, Version 3.1, Revision 5, April 2017

As follows:

> CC Part 2 conformant
>
> CC Part 3 conformant

The Common Methodology for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017 has been taken into account.

2.2 PROTECTION PROFILE CONFORMANCE CLAIM

This ST does not claim conformance of the TOE with any Protection Profile (PP).

2.3 PACKAGE CLAIM

This Security Target claims conformance to Evaluation Assurance Level 2 augmented with ALC_FLR.2 Flaw Reporting Procedures.

2.4 CONFORMANCE RATIONALE

This ST does not claim conformance of the TOE with any PP; therefore, a conformance rationale is not applicable.

3 SECURITY PROBLEM DEFINITION

3.1 THREATS

Table 3 lists the threats addressed by the TOE. Potential threat agents are unauthorized users, authorized non-administrator users, and attackers who are not TOE users. The unauthorized users are considered to possess public knowledge of how the TOE operates, and the skills and resources to alter TOE configuration settings, or parameters, or both. The unauthorized users are not granted physical or logical access to the TOE. Authorized non-administrator users are assumed to have access to the TOE, extensive knowledge of TOE operations, and to possess a high level of skill. They have moderate resources to alter TOE parameters but are assumed not to be wilfully hostile. Attackers have little knowledge of TOE operations, a low level of skill, limited resources to alter TOE parameters and no physical access to the TOE.

The assets that the TOE protects are:

1. the integrity and confidentiality of the data transmitted between the TOE and third parties,
2. the integrity of the TOE firmware,
3. the accessibility to role-protected functionality.

Mitigation to the threats is through the objectives identified in Section 4 Security Objectives.

Threat	Description
T.ACCESS	An attacker may be able to view or modify data that is transmitted between the TOE and an authorized remote external entity.
T.BAD_FW	An attacker may be able to load a not-legit TOE firmware package in the SoC microprocessor.
T.CONFIG	An authorized non-administrator user could improperly gain access to TSF functionality if the TOE is misconfigured or does not enforce proper roles and permissions.
T.UNAUTH	An unauthorized user may gain access to TOE functionally that is restricted to authorized users.

Table 3 – Threats

3.2 ORGANIZATIONAL SECURITY POLICIES

Organizational Security Policies (OSPs) are security rules, procedures, or guidelines imposed in the operational environment. Table 4 lists the OSPs that are presumed to be

imposed upon the TOE or its operational environment by an organization that implements the TOE in the Common Criteria evaluated configuration.

OSP	Description
P.CRYPTO	The TOE shall incorporate cryptographic mechanisms to protect against potential disclosure or modification of sensitive information, which is transferred between the TOE and administrators, and to verify the digital signatures of firmware packages.
P.MANAGE	The TOE shall be managed only by authorized users.

Table 4 – Organizational Security Policies

3.3 ASSUMPTIONS

The assumptions required to ensure the security of the TOE are listed in Table 5.

Assumptions	Description
A.LOCATE	The TOE will be located within controlled access facilities, which will prevent unauthorized physical access.
A.MANAGE	There are one or more competent individuals assigned to manage the TOE.

Table 5 – Assumptions

4 SECURITY OBJECTIVES

The purpose of the security objectives is to address the security concerns and to show which security concerns are addressed by the TOE, and which are addressed by the environment. Threats may be addressed by the TOE or the security environment or both. Therefore, the CC identifies two categories of security objectives:

Security objectives for the TOE.

Security objectives for the environment.

4.1 SECURITY OBJECTIVES FOR THE TOE

This section identifies and describes the security objectives that are to be addressed by the TOE.

Security Objective	Description
O.ACCESS	The TOE must allow authorized users to access only appropriate TOE functions and data.
O.AUDIT	The TOE must record time stamped audit records for use of the TOE functions. Audit records must be readable by authorized administrators and administrators must be able to filter records for ease of viewing. The TOE must also protect stored audit records. Audit records must be associated to users.
O.ADMIN	The TOE must provide all the functions and facilities necessary to support the administrators in their management of the TOE and restrict these functions and facilities from unauthorized use.
O.I&A	The TOE must be able to identify and authenticate users prior to allowing access to the administrative functions and data of the TOE.
O.PROTECT	The TOE must protect against inadvertent access to interactive management sessions and must provide a means of controlling and restricting access to TOE services and ports. The TOE must ensure the confidentiality and integrity of interactive administrative sessions.
O.FW_INTEGRITY	The TOE must provide means for verifying the integrity of the TOE firmware packages loaded in order to ensure the legitimacy of these firmware packages.

Table 6 – Security Objectives for the TOE

4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

This section identifies and describes the security objectives that are to be addressed by the IT environment or by non-technical or procedural means.

Security Objective	Description
OE.PERSONNEL	There are an appropriate number of trusted, authorized administrators trained to administer the TOE. Authorized administrators are carefully selected and trained for proper operation of the TOE, follow all administrator guidance and are not malicious.
OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from any physical attack.

Table 7 – Security Objectives for the Operational Environment

4.3 SECURITY OBJECTIVES RATIONALE

The following table maps the security objectives to the assumptions, threats, and organizational policies identified for the TOE.

	T.ACCESS	T.BAD_FW	T.CONFIG	T.UNAUTH	P.CRYPTO	P.MANAGE	A.LOCATE	A.MANAGE
O.ACCESS				X		X		
O.ADMIN			X	X		X		
O.AUDIT			X					
O.I&A			X	X				
O.PROTECT	X		X		X			
O.FW_INTEGRITY		X			X			
OE.PERSONNEL								X
OE.PHYSICAL							X	

Table 8 – Mapping Between Objectives, Threats, OSPs, and Assumptions

4.3.1 Security Objectives Rationale Related to Threats

The security objectives rationale related to threats traces the security objectives for the TOE and the Operational Environment back to the threats addressed by the TOE.

Threat:	T.ACCESS
Objectives:	O.PROTECT
Rationale:	O.PROTECT ensures that interactive administrative sessions are confidential and that they can't be modified.

Threat:	T.BAD_FW
Objectives:	O.FW_INTEGRITY
Rationale:	O.FW_INTEGRITY ensures that any TOE firmware package loaded in the TOE is verified using cryptographic means so that only legit packages with correct digital signature can be loaded.

Threat:	T.CONFIG
Objectives:	O.ADMIN
	O.AUDIT
	O.I&A
	O.PROTECT
Rationale:	O.ADMIN helps to mitigate this threat by ensuring the TOE has the proper environment in which to operate. O.AUDIT allows for the review of configuration changes thus helping to ensure that configuration changes are authorized and have been made correctly. O.I&A helps to mitigate the threat by ensuring that users are identified and authorized before they can access to TOE security functions. O.PROTECT ensures that interactive management sessions can't be inadvertently accessed and ensures that they are protected.

Threat:	T.UNAUTH

Objectives:	O.ACCESS
	O.ADMIN
	O.I&A
Rationale:	O.ACCESS helps to mitigate this threat by limiting an authorized user's access to appropriate TOE functions and data. O.I&A helps to mitigate the threat by ensuring that users are identified and authorized before they can access to TOE security functions. O.ADMIN helps to mitigate this threat by ensuring the TOE has the proper environment in which to operate.

4.3.2 Security Objectives Rationale Related to OSPs

The security objectives rationale related to OSPs traces the security objectives for the TOE back to the OSPs applicable to the TOE.

Policy:	P.CRYPTO
Objectives:	O.PROTECT
	O.FW_INTEGRITY
Rationale:	O.PROTECT ensures that the confidentiality and integrity of the TOE communications is maintained. O.FW_INTEGRITY provides cryptographic means for firmware packages digital signature verification.

Policy:	P.MANAGE
Objectives:	O.ACCESS
	O.ADMIN
Rationale:	O.ACCESS ensures that only authorized users manage the TOE. O.ADMIN ensures that the operational environment is adequate for the operation of the TOE.

4.3.3 Security Objectives Rationale Related to Assumptions

The security objectives rationale related to assumptions traces the security objectives for the operational environment back to the assumptions for the TOE's operational environment.

Assumption:	A.LOCATE
Objectives:	OE.PHYSICAL
Rationale:	OE.PHYSICAL supports this assumption by protecting the TOE from physical attack.

Assumption:	A.MANAGE
Objectives:	OE.PERSONNEL
Rationale:	OE.PERSONNEL supports this assumption by ensuring that trained individuals are in place to manage the TOE.

5 EXTENDED COMPONENTS DEFINITION

There are no extended components.

6 SECURITY REQUIREMENTS

Section 6 provides security functional and assurance requirements that must be satisfied by a compliant TOE. These requirements consist of functional components from Part 2 of the CC, and an Evaluation Assurance Level (EAL) that contains assurance components from Part 3 of the CC.

6.1 CONVENTIONS

The CC permits four types of operations to be performed on functional requirements: selection, assignment, refinement, and iteration. These operations, when performed on requirements that derive from CC Part 2, are identified in this ST in the following manner:

Selection: Indicated by surrounding brackets, e.g., [selected item].
Assignment: Indicated by surrounding brackets and italics, e.g., [*assigned item*].
Assignment within selection: Indicated by double surrounding brackets and italics, e.g., [[*assigned item within selection*]]
Refinement: Refined components are identified by using **bold** for additional information, or ~~strikeout~~ for deleted text.
Iteration: Indicated by assigning a number in parenthesis to the end of the functional component identifier as well as by modifying the functional component title to distinguish between iterations, e.g., 'FDP_ACC.1(1), Subset access control (administrators)' and 'FDP_ACC.1(2) Subset access control (devices)'.

6.2 SECURITY FUNCTIONAL REQUIREMENTS

The security functional requirements for this ST consist of the following components from Part 2 of the CC and are summarized in Table 9.

Class	Identifier	Name
Security Audit (FAU)	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_SAR.1	Audit review
	FAU_SAR.2	Restricted audit review
	FAU_SAR.3	Selectable audit review
	FAU_STG.1	Protected audit trail storage
	FAU_STG.3	Action in case of audit data loss

Cryptographic Support (FCS)	FCS_COP.1	Cryptographic operation
User Data Protection (FDP)	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute-based access control
	FDP_UCT.1	Inter-TSF user data confidentiality transfer protection
Identification and Authentication (FIA)	FIA_ATD.1	User attribute definition
	FIA_SOS.1	Verification of secrets
	FIA_UAU.2	User authentication before any action
	FIA_UID.2	User identification before any action
Security Management (FMT)	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialisation
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.1	Security roles
Protection of the TSF (FPT)	FPT_STM.1	Reliable time stamps
	FPT_RCV.3	Automated recovery without undue loss
	FPT_FLS.1	Failure with preservation of secure state
TOE Access (FTA)	FTA_SSL.1	TSF-initiated session locking (WebUI and SMASH CLP CLI)
	FTA_SSL.3	TSF-initiated termination (Redfish API)
	FTA_SSL.4	User-initiated termination
	FTA_TSE.1	TOE session establishment
Trusted path/channels (FTP)	FTP_TRP.1	Trusted path

Table 9 – Summary of Security Functional Requirements

6.2.1 Security Audit (FAU)

6.2.1.1 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) [

Login and logout
Account creation, modification, deletion, disabling, and password change
All changes to password policy
All changes to roles
Enabling and disabling of TOE system services and service port assignment changes
All changes to TOE firewall rules
KVM configuration change
TOE firmware update].

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

a) Date and time of the 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, [*interface*].

6.2.1.2 FAU_GEN.2 User Identity Association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

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

6.2.1.3 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 [*users who have been assigned the Administrator role*] with the capability to read [*all audit logs*] from the audit records.

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

6.2.1.4 FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

Dependencies: FAU_SAR.1 Audit review

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

6.2.1.5 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 [*filtering*] of audit data based on [*date*].

6.2.1.6 FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to [prevent] unauthorised modifications to the stored audit records in the audit trail.

6.2.1.7 FAU_STG.3 Action in case of audit data loss

Hierarchical to: No other components

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.3.1 The TSF shall [*replace previously saved audit trail backup file with the current audit trail file and clear all records inside the current audit trail file*] if the audit trail **file** exceeds [*400 KB in size*].

6.2.2 Cryptographic Support (FCS)

6.2.2.1 FCS_COP.1 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 The TSF shall perform [*the cryptographic operations specified in table below*] in accordance with a specified cryptographic algorithm [*the cryptographic*

algorithms specified in table below] and cryptographic key sizes [*cryptographic key sizes specified in table below*] that meet the following: [*standards listed in table below*].

Cryptographic Operation	**Algorithm**	**Key Size or Digest (bits)**	**Categorization (ACM)**	**Standard**
Encryption and Decryption	AES (CBC, CTR and GCM mode)	128, 256	CBC (R - Recommended) CTR (R - Recommended) GCM (R - Recommended)	FIPS PUB 197 (AES), NIST SP 800-38A NIST SP 800-38C NIST SP 800-38D
Cryptographic Signature Services	RSA Digital Signature Algorithm (RSASSA-PKCS-v1_5 using SHA-256 and SHA-512)	Signature Verification 1024, 2048, 3072, 4096 Signature Generation 2048, 3072, 4096	L - Legacy Note: Even the 3072 and 4096 key sizes are Recommended, they are considered as legacy due to the PKCS#1.5 mode.	PKCS #1.5
Key agreement/ Key exchange	Diffie-Hellman key agreement method	diffie-hellman-group14-256, diffie-hellman-group16-512, diffie-hellman-group18-512	Group 14 (L – Legacy 2025) Group 16 (R - Recommended) Group 18 (R - Recommended)	RFC4253 RFC4419
	EC Diffie-Hellman key agreement method	ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521	NIST P-256 (R - Recommended) NIST P-384 (R - Recommended) NIST P-521 (R - Recommended)	RFC4253 RFC5656
	ECDSA key exchange with signature verification using SHA-2	ecdsa-sha2-nistp256 with SHA-256, ecdsa-sha2-nistp384 with SHA-384, ecdsa-sha2-nistp521 with	NIST P-256 (R - Recommended) NIST P-384 (R - Recommended) NIST P-521 (R -	RFC4253 FIPS PUB 186-4

Cryptographic Operation	Algorithm	Key Size or Digest (bits)	Categorization (ACM)	Standard
		SHA-512	Recommended)	
Hashing	SHA-256	256	R - Recommended	FIPS PUB 180-4
	SHA-384	384	R - Recommended	
	SHA-512	512	R - Recommended	
Keyed Hash	HMAC-SHA-256	8 - 524288 key 256 digest	L - Legacy (2030)	FIPS PUB 198
	HMAC-SHA2-384	8 - 524288 key 384 digest	R - Recommended	
	HMAC-SHA2-512	8 - 524288 key 512 digest	R - Recommended	
Random Bit Generation	CTR_DRBG	256	R - Recommended	NIST SP800-90A

Note: ACM acronym belongs to the Agreed Cryptographic Mechanism v1.3 document (https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.3.pdf).

6.2.3 User Data Protection (FDP)

6.2.3.1 FDP_UCT.1 Inter-TSF user data confidentiality transfer protection

Hierarchical to: No other components.

Dependencies: [FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
[FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_UCT.1.1 The TSF shall enforce the [*Security Management Access Control SFP*] to [transmit and receive] user data in a manner protected from unauthorized disclosure.

6.2.3.2 FDP_ACC.1 Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the [*Security Management Access Control SFP*] on [

a) Subjects: authorized users
b) Objects: TOE configuration

c) Operations: view, modify

].

6.2.3.3 FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 The TSF shall enforce the [*Security Management Access Control SFP*] to objects based on the following: [

a) Subjects: authorized users
b) Subject attributes: role and associated permissions
c) Objects: TOE configuration
d) Object attributes: none].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*users can perform the actions determined by the user's role and the role's permissions.*].

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

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

6.2.4 Identification and Authentication (FIA)

6.2.4.1 FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies.

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

a) Username
b) User roles

].

6.2.4.2 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 [

a) a configurable minimum length of 8 to 16 characters,
b) passwords must contain at least three of the following character types:
uppercase letters,
lowercase letters,
numbers, and

special characters

c) configurable number of historical passwords must not be reusable].

6.2.4.3 FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification

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

6.2.4.4 FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

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

6.2.5 Security Management (FMT)

6.2.5.1 FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

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

FMT_MSA.1.1 The TSF shall enforce the [*Security Management Access Control SFP*] to restrict the ability to [query, modify, delete] the security attributes [*user role*] to [*users who have been assigned the Administrator role*].

6.2.5.2 FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

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

FMT_MSA.3.1 The TSF shall enforce the [*Security Management Access Control SFP*] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [*users who have been assigned the Administrator role*] to specify alternative initial values to override the default values when an object or information is created.

6.2.5.3 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) User management
b) Service settings
c) Firewall settings
d) Audit management
e) Power control
f) Remote control
g) System maintenance

].

6.2.5.4 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 [*Administrator, Operator, User*].

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

6.2.6 Protection of the TSF (FPT)

6.2.6.1 FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

Dependencies: No dependencies.

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

6.2.6.2 FPT_RCV.3 Automated recovery without undue loss

Hierarchical to: FPT_RCV.2 Automated recovery

Dependencies: AGD_OPE.1 Operational user guidance

FPT_RCV.3.1 When automated recovery from [*FW package integrity failure*] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2 For [*FW package integrity failure*], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3 The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding [*missing default configuration*] for loss of TSF data or objects under the control of the TSF.

FPT_RCV.3.4 The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.

6.2.6.3 FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

Dependencies: No other components.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur [*fail in the FW package integrity check*].

Application note:

The failure in the FW integrity check can happen either during boot or during a new FW load. In both cases the TOE will:

- √ Discard the FW causing the integrity error and notify the administrator if he/she is in a firmware update process and terminate the process
- √ Log the event
- √ Reload the FW from the memory-source-2 which stores a backup of the original FW.

6.2.7 TOA Access (FTA)

6.2.7.1 FTA_SSL.1 TSF-initiated session locking (WebUI and SMASH CLP CLI)

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL.1.1 The TSF shall lock an interactive session after [*a time interval of user inactivity that has been configured by a user with the Administrator or Operator role using the WebUI or Redfish API*] by:

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

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

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session: [*re-authentication*].

6.2.7.2 FTA_SSL.3 TSF-initiated termination (Redfish API)

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [*time interval of user inactivity that has been configured by the session owner during session establishment or based on a global setting configured by a user with the Administrator or Operator role*].

6.2.7.3 FTA_SSL.4 User-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_SSL.4.1 The TSF shall allow user-initiated termination of the user's own interactive session.

6.2.7.4 FTA_TSE.1 TOE session establishment

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [*session establishment request source IP, source MAC address, source port number and destination port number*].

6.2.8 Trusted Path/Channels (FTP)

6.2.8.1 FTP_TRP.1 Trusted path

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [remote] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [modification, disclosure].

FTP_TRP.1.2 The TSF shall permit [remote users] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [[*administration of the TOE*]].

6.3 SECURITY ASSURANCE REQUIREMENTS

The assurance requirements are summarized in Table 10.

Assurance Class	Assurance Components	
	Identifier	**Name**
Development (ADV)	ADV_ARC.1	Security architecture description
	ADV_FSP.2	Security-enforcing functional specification
	ADV_TDS.1	Basic design
Guidance Documents (AGD)	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-Cycle Support (ALC)	ALC_CMC.2	Use of a CM system
	ALC_CMS.2	Parts of the TOE CM coverage

Assurance Class	Assurance Components	
	Identifier	**Name**
	ALC_DEL.1	Delivery procedures
	ALC_FLR.2	Flaw reporting procedures
Security Target Evaluation (ASE)	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
Tests (ATE)	ATE_COV.1	Evidence of coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
Vulnerability Assessment (AVA)	AVA_VAN.2	Vulnerability analysis

Table 10 – Security Assurance Requirements

6.4 SECURITY REQUIREMENTS RATIONALE

6.4.1 Security Functional Requirements Rationale

The following Table provides a mapping between the SFRs and Security Objectives.

	O.ACCESS	O.FW_INTEGRITY	O.ADMIN	O.AUDIT	O.I&A	O.PROTECT
FAU_GEN.1				X		
FAU_GEN.2				X		
FAU_SAR.1				X		
FAU_SAR.2				X		
FAU_SAR.3				X		
FAU_STG.1				X		
FAU_STG.3				X		
FCS_COP.1						X
FDP_ACC.1			X			
FDP_ACF.1			X			
FDP_UCT.1			X			
FIA_ATD.1	X					
FIA_SOS.1					X	
FIA_UAU.2	X				X	
FIA_UID.2	X				X	
FMT_MSA.1			X			
FMT_MSA.3			X			
FMT_SMF.1			X			
FMT_SMR.1			X			
FPT_STM.1				X		
FPT_RCV.3		X				
FPT_FLS.1		X				

	O.ACCESS	O.FW_INTEGRITY	O.ADMIN	O.AUDIT	O.I&A	O.PROTECT
FTA_SSL.1						X
FTA_SSL.3						X
FTA_SSL.4						X
FTA_TSE.1						X
FTP_TRP.1						X

Table 11 – Mapping of SFRs to Security Objectives

6.4.2 SFR Rationale Related to Security Objectives

The following rationale traces each SFR back to the Security Objectives for the TOE.

Objective:	O.ACCESS	
Security Functional Requirements:	FIA_UAU.2	User authentication before any action
	FIA_UID.2	User identification before any action
	FIA_ATD.1	User attribute definition
Rationale:	FIA_UAU.2 requires that any user be authenticated prior to being able to access TOE functionality. FIA_UID.2 requires that any user be identified prior to being able to access TOE functionality. FIA_ATD.1 states that the TSF maintains a list of user attributes that allow or deny access to TOE functionality.	

Objective:	O.FW_INTEGRITY	
Security Functional Requirements:	FPT_RCV.3	Automated recovery without undue loss
	FPT_FLS.1	Failure with preservation of secure state
Rationale:	FPT_RCV.3 provides automated recovery means for ensuring that the TOE	

	integrity remains right after detecting a failure in the verification, by ensuring the rollback to a previous secure state of operation. FPT_FLS.1 contributes to O.FW_INTEGRITY by ensuring that, in the event that the firmware is corrupt, the TOE runs the preliminary firmware and its secure state.

Objective:	O.ADMIN	
Security Functional Requirements:	FMT_MSA.1	Management of security attributes
	FMT_MSA.3	Static attribute initialisation
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute-based access control
	FDP_UCT.1	Inter-TSF user data confidentiality transfer protection
Rationale:	FMT_MSA.1 and FMT_MSA.3 restrict specified management activities to users assigned the Administrator role and ensure that appropriate default values are used. FMT_SMF.1 defines the management activities that an authorized user can perform. FMT_SMR.1 defines the three user roles which are Administrator, Operator, and User. FDP_ACC.1 enforces the security functional policy imposed on specific objects and roles. FDP_ACF.1 the security functional policy is enforced by the TSF. It explicitly allows access of TOE security functionality to authorized users and denies access to unauthorized users. FDP_UCT.1 contributes to O.ADMIN by ensuring that the data transmitted between the TOE parts remains protected and therefore any administrative action makes use of legit user data.	

Objective:	O.AUDIT

Security Functional Requirements:	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_SAR.1	Audit review
	FAU_SAR.2	Restricted audit review
	FAU_SAR.3	Selected Audit review
	FAU_STG.1	Protected audit trail storage
	FAU_STG.3	Action in case of audit data loss
	FPT_STM.1	Reliable time stamps
Rationale:	FAU_GEN.1 outlines what data must be included in audit records and what events must be audited. FAU_GEN.2 associates user information to the audit records. FAU_SAR.1 allows the administrator to view audit events. FAU_SAR.2 restricts the audit log review to users who have been assigned the Administrator role. FAU_SAR.3 allows a user who has been assigned the Administrator role to search the audit logs using filters. FAU_STG.1 does not allow unauthorised modifications or deletions of the audit logs. FAU_STG.3 saves a copy of the audit logs when the size limit is reached. This copy overwrites the previously saved copy. This ensures that recent audit data is preserved. FPT_STM.1 ensures that there is a time stamp for the audit records.	

Objective:	O.I&A	
Security Functional Requirements:	FIA_SOS.1	Specification of secrets
	FIA_UAU.2	User authentication before any action
	FIA_UID.2	User identification before any action
Rationale:	FIA_SOS.1 specifies the password rules that are enforced by the TOE. FIA_UAU.2 requires that any user be authenticated prior to being able to access TOE functionality. FIA_UID.2 requires that any user be identified prior to being able to access TOE functionality.	

Objective:	O.PROTECT	
Security Functional Requirements:	FCS_COP.1	Cryptographic operation
	FTA_SSL.1	TSF-initiated session locking (WebUI and SMASH CLP CLI)
	FTA_SSL.3	TSF-initiated termination (Redfish API)
	FTA_SSL.4	User-initiated termination
	FTA_TSE.1	TOE session establishment
	FTP_TRP.1	Trusted Path
Rationale:	FCS_COP.1 ensures that the TOE uses validated cryptography. FTA_SSL.1, FTA_SSL.3 and FTA_SSL.4 ensure that inactive sessions automatically lock and that users can logout. FTP_TRP.1 protects the management sessions from disclosure using TLS v1.2 and v1.3. FTA_TSE.1 ensures that interactive management sessions can be restricted based on origin and destination information.	

6.4.3 Dependency Rationale

Table 12 identifies the Security Functional Requirements from Part 2 of the CC and their associated dependencies. It also indicates whether the ST explicitly addresses each dependency.

SFR	Dependency	Dependency Satisfied	Rationale
FAU_GEN.1	FPT_STM.1	FPT_STM.1	
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1	
	FIA_UID.1	FIA_UID.2	FIA_UID.2 is hierarchical to FIA_UID.1; this dependency has been satisfied.
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1	
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1	
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1	

SFR	Dependency	Dependency Satisfied	Rationale
FAU_STG.1	FAU_GEN.1	FAU_GEN.1	
FAU_STG.3	FAU_STG.1	FAU_STG.1	
FCS_COP.1	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Justified	There is no need to generate the keys, they are generated and configured in the TOE during the installation process. The TOE does not generate keys in its normal operation mode.
	FCS_CKM.4	Justified	The TOE does not delete keys. The keys are protected and used during the normal operation of the TOE. As the keys used are not changing, they are always needed, so deleting them is not possible.
FDP_ACC.1	FDP_ACF.1	FDP_ACF.1	
FDP_ACF.1	FDP_ACC.1	FDP_ACC.1	
	FMT_MSA.3	FMT_MSA.3	
FDP_UCT.1	FTP_ITC.1 or FTP_TRP.1	FTP_TRP.1	
	FDP_ACC.1 or FDP_IFC.1	FDP_ACC.1	
FIA_ATD.1	None	N/A	
FIA_SOS.1	None	N/A	
FIA_UAU.2	FIA_UID.1	FIA_UID.2	FIA_UID.2 is hierarchical to FIA_UID.1; this dependency has been satisfied.
FIA_UID.2	None	N/A	
FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1	FDP_ACC.1	
	FMT_SMR.1	FMT_SMR.1	

SFR	Dependency	Dependency Satisfied	Rationale
	FMT_SMF.1	FMT_SMF.1	
FMT_MSA.3	FMT_MSA.1	FMT_MSA.1	
	FMT_SMR.1	FMT_SMR.1	
FMT_SMF.1	None	N/A	
FMT_SMR.1	FIA_UID.1	FIA_UID.2	FIA_UID.2 is hierarchical to FIA_UID.1; this dependency has been satisfied.
FPT_STM.1	None	N/A	
FPT_RCV.3	AGD_OPE.1	AGD_OPE.1	
FTA_SSL.1	FIA_UAU.1	FIA_UID.2	FIA_UAU.2 is hierarchical to FIA_UAU.1; this dependency has been satisfied.
FTA_SSL.3	None	N/A	
FTA_SSL.4	None	N/A	
FTA_TSE.1	None	N/A	
FTP_TRP.1	None	N/A	

Table 12 – Functional Requirement Dependencies

6.4.4 Security Assurance Requirements Rationale

The TOE assurance requirements for this ST consist of the requirements corresponding to the EAL 2 level of assurance, as defined in the CC Part 3, augmented by the inclusion of Flaw reporting procedures (ALC_FLR.2). EAL 2 was chosen for competitive reasons. The developer is claiming the augmentation since there are several areas where current practices and procedures exceed the minimum requirements for EAL 2.

7 TOE SUMMARY SPECIFICATION

This section provides a description of the security functions and assurance measures of the TOE that meet the TOE security requirements.

7.1 SECURITY AUDIT

The TOE locally records the audit logs of various configuration and management operations. The logs include the following events:

Login and logout

Account creation, modification, deletion, disabling, and password change

All changes to password policy

All changes to roles

Enabling and disabling of TOE system services and service port assignment changes

All changes to TOE firewall rules

KVM configuration change

TOE firmware update.

Each event contains the following information if is applicable: user identification, date/time, software interface, username, IP address or hardware interface, and event description (event type, event information, event outcome).

A user with the Administrator role who logs in to TOE Web User Interface, SMASH CLP CLI, or uses the Redfish API can read all audit information and can query audit information. Logs can be filtered by date. Audit log records are protected from modification.

The audit log is stored in the flash storage media located in the microprocessor and unaffected by system power loss. 400KB disk space is dedicated to the storage of current Audit Log file. When TOE detects that the Audit log file is reaching 400K in size, the file will be saved as a backup Audit log file and a new empty Audit log file will be used to store the upcoming log records. The previously saved backup Audit log file is removed from the storage each time a new backup file is created. The backup audit log file can be downloaded through the WebUI.

TOE Security Functional Requirements addressed: FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_STG.1, FAU_STG.3.

7.2 CRYPTOGRAPHIC SUPPORT

The TOE uses scheme approved cryptography to protect the communication links between the TOE and its remote users, and for the verification of the firmware digital

signature. Cryptographic operations are performed in accordance with the detail provided in Section 6.2.2.1.

TOE Security Functional Requirements addressed: FCS_COP.1.

7.3 USER DATA PROTECTION

The TOE provides controlled access to the administrative functions that support the remote management functionality, including:

User management

Service settings

Firewall settings

Audit management

Power control

Remote control

System maintenance

Access to these functions is controlled through the security management access control SFP, which allows users to perform functions according to assigned roles. The communication for the remote management is protected from disclosure.

The mapping between the roles and permissions are shown in the following table. These apply to WebUI and Redfish API users unless otherwise indicated.

Role	Permission
Administrator	Full access to User management including, o Account creation, modification, deletion, disablement, and password change o Password policy management; and, o Role and Privilege Management Service settings Firewall settings Audit management (Query of audit records only) Power control (Host) Remote control (Configure and Access Host

Role	Permission
	KVM) System maintenance
Operator	Full access to User management including: o Read access to role and privilege management Service settings Firewall settings Power control（Power on） (Host) Remote control (Configure and Access Host KVM) System maintenance
User	Read Access to, Role and Privilege Management only) Service settings Firewall settings Power control (Host Power Status) (Host) Remote control (Access of Host KVM) System maintenance

Table 13 - Mapping Between Roles and Permissions

TOE Security Functional Requirements addressed: FDP_ACC.1, FDP_ACF.1, FDP_UCT.1.

7.4 IDENTIFICATION AND AUTHENTICATION

The TOE supports user identification and authentication based on username and password. The TSF does not allow any TSF mediated actions before a user has been authenticated.

The TOE enforces password complexity that is configured by the administrator. The minimum password length can be configured to be from 8 to 16 characters and the character type can be set to require three or more arbitrary combinations of uppercase letters, lowercase letters, numbers, and special characters.

TOE Security Functional Requirements addressed: FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UID.2.

7.5 SECURITY MANAGEMENT

TOE provides three default user roles, and they are Administrator, Operator and User which are assigned to user accounts. The TOE has a default Administrator role account which cannot be deleted. A newly created user account is automatically assigned the user role. All accounts can only be created or modified by a user who has been assigned the Administrator role and these users are responsible for:

User management using the WebUI or Redfish API consisting of account management, password policy management, role management, and privilege management.

TOE system service management using the WebUI or Redfish API

TOE firewall rule management using the WebUI

Reviewing audit records using the WebUI, Redfish API, or SMASH CLP CLI

Power control of the host using the WebUI or Redfish API

Remote control and management of the host using the WebUI or Redfish API

System maintenance of the host and TOE using the WebUI or Redfish API

Security Management Function	Details
System Service Management	Network service ports and ports using insecure protocols and unused network service ports are closed.
Firewall Rule Management	Firewall rules can be created to filter network traffic based on IP address, port, and MAC address.
Remote Control and Management of the Host	Remote Control redirects the console of the server system to the connected WebUI or Redfish API session allowing the remote viewing of the host's display and control of the host's keyboard/mouse.
System Maintenance of the Host	The host's firmware can be updated using the WebUI and Redfish API.

Table 14 - System Management Function Details

TOE Security Functional Requirements addressed: FMT_MSA.1, FMT_MSA.3, FMT_SMF.1, FMT_SMR.1.

7.6 PROTECTION OF THE TSF

The TOE provides reliable time stamp services for the TOE audit function and TSF initiated session locking using the system clock in the BIOS. The time settings are only accessible by the administrator role.

The integrity and authenticity of the TOE FW is protected based on firmware root of trust (ROT), which contains immutable trust data (consists on a root public key that is hardcoded and cannot be modified nor deleted). In this sense, the OTP (One Time Programmable) area of BMC chip is used as ROT.

The TOE FW is digitally signed before publishing to ensure FW integrity and authenticity. The hash value of the TOE FW is computed first, and then is encrypted using the private key of the key pair. The encrypted value is the digital signature value of the FW, which is compiled together with the FW into the image file. When updating the TOE FW, the public key of the preceding key pair is used to decrypt the encrypted value and compare the hash values to ensure the authenticity and integrity of the FW. If the public key is changed or tampered with, the FW authentication will fail. The TOE uses a read-only area of microprocessor chip to store the public key, that is, the public key is stored in it, to protect the public key from tampering or counterfeiting. To perform signature verification on the TOE FW, it reads the public key from the chip. FW booting must not be executed unless passing verification based on these data.

During TOE booting, the first FW package of the TOE, will verify the authenticity and integrity of the next FW package by calculating and checking hash value of the public key inside the next package, and verifying the digital signature of it by this authenticated public key. This next FW package will be executed only after successful verification, and it will load and verify the following FW packages in the similar way by using public keys which is already verified by the former FW package. Therefore, the integrity and authenticity of the FW is assured.

During TOE updating, the FW package of the TOE will verify the authenticity and integrity of the FW package which be updated by calculating and checking hash value of the public key, and verifying the digital signature of it by this public key. This new FW package will be executed only after successful verification. Therefore, the integrity and authenticity of the FW is assured. It will not perform updating if the verification is failed.

Both updating and booting operation are atomic. The TOE preserves a secure state if any failure occurs during booting or updating, and will not enter into operation unless authenticity and integrity verification to FW package is successful. If the TOE is failed to restore to a former configuration (e.g. due to file damage) after reboot, it will be restart from backup FW to maintain security.

TOE Security Functional Requirements addressed: FPT_STM.1, FPT_RCV.3, FPT_FLS.1.

7.7 TOE ACCESS

Remote Management of the TOE are provided through the WebUI, Redfish API, and SMASH CLP CLI interfaces. After the configured session timeout has passed, the WebUI, and SMASH CLP CLI user sessions will become invalid, and the system will automatically terminate the connection. The user needs to log in again in order to perform any operations. The session timeout can be configured by a user with the Administrator or Operator role using the WebUI or Redfish API interfaces.

Upon successful authentication at the Redfish API interface, a X-Auth-Token is issued to the user. The X-Auth-Token can be attached to subsequent Redfish API requests from the same user, replacing user credentials, allowing those requests to be executed with the same user privilege. A user with the Administrator or Operator role can configure a session timeout. Once the configured timeout has passed, the X-Auth-Token is no longer accepted by the TOE and new Redfish API requests issued with the X-Auth-Token will be rejected. The user needs to provide username and password again at the Redfish API interface to obtain a new X-Auth-Token. All Redfish API users have the option to specify an alternative session time out value during the initial authentication request to overwrite the administrator configured X-Auth-Token session timeout. Alternatively, Redfish API users may choose to use basic HTTP authentication instead. In this case, the user provides credentials with each Redfish API request, and each Redfish API request is regarded as an individual user session.

The session can be terminated in either of the following ways:

1. Termination upon timeout: If a web, or SSH session is inactive until the timeout period expires, the session is automatically disconnected.

2. Manual termination: The TOE allows users to actively end sessions. After the session ends, the user will need to log in again to perform any TOE operations.

The TOE allows users with the Administrator or Operator role to disable services to ensure that only the services being used are available. Users assigned to the Administrator or Operator can configure the port number on which the service is available.

Additionally, the TOE supports connection control based on source IP address, port, and MAC address. Rules can be configured by a user with the Administrator or Operator role to allow or deny connections from corresponding sources.

TOE Security Functional Requirements addressed: FTA_SSL.1, FTA_SSL.3, FTA_SSL.4, FTA_TSE.1.

7.8 TRUSTED PATH / CHANNELS

When the TOE Web User Interface or Redfish API interface is used, the connection between TOE and the remote user's session is protected from modification and disclosure using TLS v1.2 and v1.3 (RFC5246 and RFC8446 respectively). The SMASH CLP

CLI is protected by SSH v2 (RFCs 4253, 6668, 8268, 8308, 8332, 8709, 8758, 9142). These connections are logically distinct from other communication channels.

The cipher suites that the TOE accepts are:

TLS v1.3 (R - Recommended)

TLS_AES_256_GCM_SHA384 (R - Recommended)

TLS_AES_128_GCM_SHA256 (R - Recommended)

TLS v1.2 (R - Recommended)

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (L – Legacy)

TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (L – Legacy)

TLS_DHE_RSA_WITH_AES_256_CCM (L – Legacy)

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (L – Legacy)

TLS_DHE_RSA_WITH_AES_128_CCM (L – Legacy)

On the other hand, the certificate installed by default is used to connect to the TOE through TLS channel. The TLS settings functionality allows to the administrator to view/modify the certificate. If the administrator modifies the certificate (should be the pem type), the TOE will not be covered by the TOE evaluated configuration.

The protection of the SSH channel available in the TOE is the following:

kex_algorithms:

- diffie-hellman-group14-sha256
- diffie-hellman-group16-sha512
- diffie-hellman-group18-sha512
- ecdh-sha2-nistp521
- ecdh-sha2-nistp384
- ecdh-sha2-nistp256

server_host_key_algorithms:

- rsa-sha2-512
- rsa-sha2-256
- ecdsa-sha2-nistp521
- ecdsa-sha2-nistp384
- ecdsa-sha2-nistp256

encryption_algorithms:

- aes256-ctr
- aes128-ctr

mac_algorithms:

- hmac-sha2-512
- hmac-sha2-256

TOE Security Functional Requirements addressed: FTC_TRP.1.

8 TERMINOLOGY AND ACRONYMS

8.1 TERMINOLOGY

The following terminology is used in this ST:

Term	Description
Security Policy	The term security policy is used in this ST to describe the policies implemented within the TOE to enforce the claimed functionality. It does not refer to the specific policies enforced by the User Data Protection SFRs.

Table 15 – Terminology

8.2 ACRONYMS

The following acronyms are used in this ST:

Acronym	Definition
ACM	Agreed Cryptographic Mechanisms
CC	Common Criteria
CLI	Command Line Interface
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
HTTPS	Hypertext Transfer Protocol Secure
IPMI	Intelligent Platform Management Interface
NTP	Network Time Protocol
PP	Protection Profile
SFP	Security Function Policy
SFR	Security Functional Requirement
SMASH CLP	System Management Architecture for Server Hardware Command Line Protocol
SNMP	Secure Network Mail Protocol
SSH	Secure Shell
ST	Security Target

Acronym	Definition
TCP	Transmission Control Protocol
TOE	Target of Evaluation
TSF	TOE Security Functionality

Table 16 – Acronyms