

Security Target of Huawei iBMC

Issue 2.0
Date 2022-10-29

HUAWEI TECHNOLOGIES CO., LTD



Copyright © Huawei Technologies Co., Ltd. 2017. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided “AS IS” without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Huawei Technologies Co., Ltd.

Address: Huawei Industrial Base
Bantian, Longgang
Shenzhen 518129
People’s Republic of China

Website: <http://www.huawei.com>

Email: support@huawei.com

Contents

| | |
|--|-----------|
| 1 Introduction..... | 8 |
| 1.1 ST Identification..... | 8 |
| 1.2 TOE Identification..... | 8 |
| 1.3 TOE overview | 8 |
| 1.4 TOE Description..... | 11 |
| 1.4.1 Physical Scope of the TOE..... | 11 |
| 1.4.2 Logical scope of the TOE..... | 11 |
| 1.4.3 Non-TOE hardware and software | 12 |
| 1.4.4 Summary of Security Features | 13 |
| 2 CC Conformance Claims | 16 |
| 2.1 CC Conformance Claim..... | 16 |
| 3 TOE Security Problem Definition | 17 |
| 3.1 Asset Definitions | 17 |
| 3.2 Threats to the Assets Definition..... | 17 |
| 3.3 Organizational Security Policies | 19 |
| 3.3.1 P.ChangeForced | 19 |
| 3.3.2 P.Ctlr_FWU..... | 20 |
| 3.4 Assumptions..... | 20 |
| 3.4.1 A.PhysicalProtection..... | 20 |
| 3.4.2 A.ManagementNetwork | 20 |
| 3.4.3 A.InternalConnections | 20 |
| 3.4.4 A.NetworkSegregation..... | 20 |
| 3.4.5 A.TrustedPersonnel..... | 21 |
| 3.4.6 A.HardwareROT..... | 21 |
| 3.4.7 A.ResponsibleUser | 21 |
| 3.4.8 A.ProtectedUpdate..... | 21 |
| 4 Security Objectives..... | 22 |
| 4.1 Security objectives for the TOE | 22 |
| 4.1.1 O.Authentication | 22 |
| 4.1.2 O.Authorization..... | 22 |
| 4.1.3 O.FWIntegrity | 23 |
| 4.1.4 O.FWValidation..... | 23 |

| | |
|---|-----------|
| 4.1.5 O.Communication | 23 |
| 4.1.6 O.Audit | 23 |
| 4.1.7 O.SecurityManagement | 24 |
| 4.1.8 O.LifeCycle..... | 24 |
| 4.1.9 O.Ctrl_FWU..... | 25 |
| 4.2 Security Objectives for the Operational Environment | 25 |
| 4.2.1 OE.ExclusiveProtHost | 25 |
| 4.2.2 OE.CorrectHosts..... | 25 |
| 4.2.3 OE.PhysicalProtection | 26 |
| 4.2.4 OE.ManagementNetwork..... | 26 |
| 4.2.5 OE.NetworkSegregation | 26 |
| 4.2.6 OE.TrustedPersonnel | 26 |
| 4.2.7 OE.CredentialProtection | 27 |
| 4.2.8 OE.AuditStorage | 27 |
| 4.2.9 OE.FWU_Usage..... | 27 |
| 4.2.10 OE.ImmutableROT..... | 27 |
| 4.2.11 OE. IntegritybyBootROM | 28 |
| 4.3 Overview: assumptions, policies and objectives for the environment | 28 |
| 4.4 Threats and Security Objectives Rationale..... | 29 |
| 4.4.1 Threats and Security Objective Rational for the base Package..... | 29 |
| 4.4.2 Threats and Security Objective Rational for the FWU package | 32 |
| 4.5 Assumption, Polices and Objective for the Environment Rationale | 32 |
| 4.5.1 A, P and OE Rationale for the base package | 32 |
| 4.5.2 A, P and OE Rationale for the FWU package..... | 34 |
| 5 New and Extended Components Definition | 36 |
| 5.1 Extended Functional Components Definition | 36 |
| 5.1.1 FPT_ARB Anti-Rollback..... | 36 |
| 5.1.2 Extended Family FPT_FLS Failure with preservation of secure state | 37 |
| 5.1.3 FPT_ROTSTB Root of trust secure booting..... | 39 |
| 5.1.4 FPT_ROTSTU Root of trust secure updating..... | 41 |
| 5.1.5 Extended FPT_TST_EXT.1 TSF testing..... | 43 |
| 5.2 Extended Assurance Components Definition..... | 45 |
| 6 Security Requirements for the TOE | 46 |
| 6.1 Conventions | 46 |
| 6.2 Security Functional Requirements..... | 46 |
| 6.2.1 Security Audit (FAU)..... | 46 |
| 6.2.2 Cryptographic Support (FCS)..... | 48 |
| 6.2.3 User Data Protection (FDP) | 48 |
| 6.2.4 Identification and Authentication (FIA)..... | 49 |
| 6.2.5 Security Management (FMT)..... | 51 |
| 6.2.6 Protection of the TSF (FPT)..... | 52 |

| | |
|---|-----------|
| 6.2.7 TOE access (FTA) | 57 |
| 6.2.8 Trusted Path/Channel (FTP)..... | 58 |
| 6.2.9 FRU_FLT.1 Degraded fault tolerance | 58 |
| 6.3 TOE's FWU package..... | 59 |
| 6.4 Security Functional Requirements Rationale | 60 |
| 6.4.1 Objectives and Security Functional Requirements | 60 |
| 6.4.2 Security Functional Requirements Dependency Rationale | 61 |
| 6.4.3 Justification for SFRs covering O.Authentication | 68 |
| 6.4.4 Justification for the SFRs covering O.Authorization | 68 |
| 6.4.5 Justification for the SFRs covering O.FWIntegrity | 68 |
| 6.4.6 Justification for the SFRs covering O.FWValidation | 69 |
| 6.4.7 Justification for the SFRs covering O.Communication..... | 70 |
| 6.4.8 Justification for the SFRs covering O.Audit..... | 70 |
| 6.4.9 Justification for the SFRs covering O.SecurityManagement..... | 70 |
| 6.4.10 Justification for the SFRs covering O.LifeCycle | 71 |
| 6.4.11 Justification for the SFRs added by O.Ctrl_FWU..... | 71 |
| 6.5 Security Assurance Requirements | 72 |
| 6.6 Security Assurance Requirements Rationale..... | 72 |
| 7 TOE Summary Specification | 73 |
| 7.1 Security Audit | 73 |
| 7.2 Cryptographic functions | 74 |
| 7.3 Identification and Authentication | 74 |
| 7.4 Security Management | 75 |
| 7.5 Protection of the TSF..... | 76 |
| 7.6 TOE Access..... | 77 |
| 7.7 Trusted Path/Channel, User Data Protection | 77 |
| A Abbreviations, Terminology and References..... | 80 |
| A.1 Abbreviations | 80 |
| A.2 Terminology | 81 |
| A.3 References..... | 81 |

Figures

Figure 1-1 Operational environment of the TOE 10

Tables

| | |
|--|----|
| Table 1-1 Supported Huawei server models..... | 9 |
| Table 1-2 TOE Evaluated Configuration..... | 11 |
| Table 6-1 User roles | 52 |
| Table 7-1 Groups of accounts..... | 75 |

1 Introduction

1.1 ST Identification

Title: Security Target of Huawei iBMC

Version: 2.0

Date: 2022-10-29

Developer: Huawei Technologies Co., Ltd.

1.2 TOE Identification

Name: Huawei Server intelligent Baseboard Management Controller iBMC

Version: 3.03.00.21

Developer: Huawei Technologies Co., Ltd.

1.3 TOE overview

1.1.1 TOE type and usage

Huawei Server intelligent Baseboard Management Controller iBMC is an out of band management system optimized for server remote management. It consists of hardware and firmware embedded on the motherboard of Huawei server, and independent to in-band components such as host processors, OS, storages, etc., so that the server can be managed without relying on the status of the in-band components, and the management operation will not interfere with services run on the server.

In the further, following abbreviations are used in this ST: FW means firmware, HW means hardware and SW means software. The FW of iBMC is called Target of Evaluation (TOE).

The TOE provides following server management functionalities with abundant management interfaces both for human being and machine users:

- ✓ Hardware monitoring
- ✓ Fault diagnostics and management
- ✓ Energy management

- ✓ Domain management and directory service
- ✓ Firmware management
- ✓ Storage management
- ✓ Server remote control
- ✓ Firmware(e.g. BIOS) updating

To facilitate all kinds of IT system integration, the TOE provides plentiful management interfaces including: CLI, WebUI, IPMI, SNMP, and Redfish API. Generally, CLI and WebUI are interfaces for human being users, while IPMI, SNMP and Redfish API are interfaces for machine users. All the user interfaces use authentication mechanisms and encryption algorithms to protect access and transmission security. Though both IPMI and SNMP protocol were very popular in many years, as they support very limited security algorithms which may not very strong, Redfish API is recommended for machine-machine interfaces, and the certified configuration in this ST doesn't include IPMI and SNMP. Please refer to chapter1.4.2 for detailed logical scope of the TOE.

1.1.2 TOE operational environment

As shown in the Figure1-1, a Huawei server consists of two logical plane:

- Management plane: provides server management functionalities
- Service plane: runs services and applications that a server is expected to, e.g. web server, DB server, etc.

The iBMC constitutes thereby the management plane which is hosted by the server, yet the server itself including its service plane is out of the TOE scope. The iBMC consists of firmware which is also the TOE, and hardware (a SOC with its BootROM program, an EMMC flash) as part of operational environment on which the TOE runs.

The service plane is consists of all in-band components including CPU, memories, hard disks, host OS, etc. The TOE accesses, monitors and manages the service plane via PCIe (Peripheral Component Interconnect Express), USB(Universal Serial Bus) and LPC(Low Pin Count), which are physical interfaces inside server and are all physically protected by operational environment.

Following are supported Huawei server models that host TOE:

Table 1-1 Supported Huawei server models

| Category | Supported Servers |
|-----------------|--|
| TaiShan Servers | TaiShan 200 2280, TaiShan 200 5280, TaiShan 200 2180, TaiShan 200 1280, TaiShan 200 2480, TaiShan 200 5290, TaiShan 200 2280E, TaiShan 200 2280M, TaiShan 200K 2180K, TaiShan 200K 2480K, TaiShan 200K 2280K, TaiShan 200K 5280K, TaiShan 200 Pro 2480, TaiShan 200 Pro 2280 |
| KunPeng Servers | S920X00, S920X01, S920S00, S920S01, S920X00K, S920X01K, S920S00K, S920S01K, S920X02, S920X02K, S920X03, S920S03, S920X05 |
| Atlas Servers | Atlas 800 9000, Atlas 800 9010, Atlas 500 Pro 3000, Atlas 800 3000 |
| Storage | OceanStor Pacific 9550, OceanStor Pacific 9950 |
| TCE | E9000H-4, E9000H-8 |

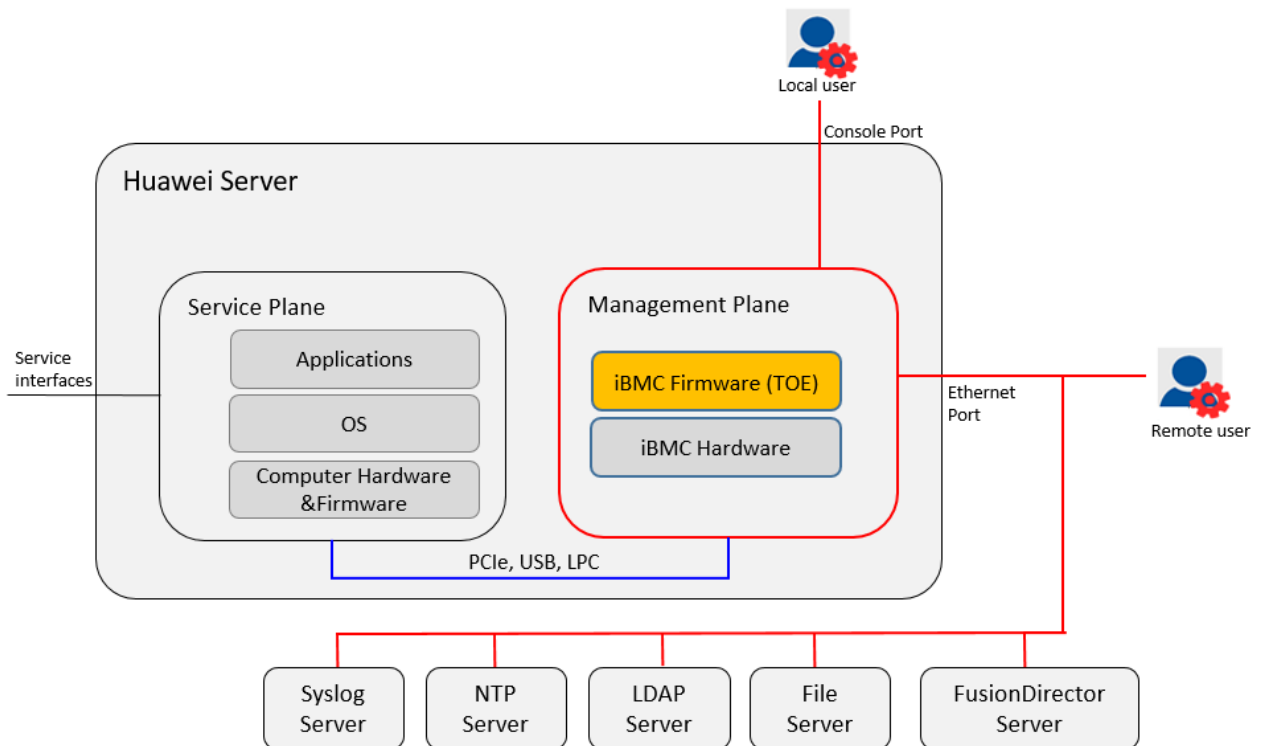


Figure 1-1 Operational environment of the TOE

The red lines in the Figure 1-1 represent local management sub-network, which shall be separated with other service networks and not connected to internet. It is assumed that the TOE together with all local management devices are deployed and operated in a physical secure environment that prevent unauthorized or unauthenticated physical accessing to the TOE and its underlying hardware.

To perform all functionalities, the TOE cooperated with these devices deployed in the local management sub-network as below:

- ✓ Syslog server for collecting and managing logs
- ✓ NTP server for synchronizing system time
- ✓ LDAP server for integration of authentication and authorization
- ✓ File server for preserving updating images of the TOE
- ✓ FusionDirector Server for centralized management of multiple TOEs

Remote TOE users can access the TOE through web browser, SSH client, while local user can access directly the TOE by physically connecting the console port. Both them are authenticated and authorized before any action.

1.4 TOE Description

1.4.1 Physical Scope of the TOE

The evaluated configuration of the TOE is listed as below.

Table 1-2 TOE Evaluated Configuration

| Type | Delivery Item | Version | Format | Delivery method |
|------------------|---|------------|----------------|-------------------------|
| Firmware | iBMC 3.03.00.21 | 3.03.00.21 | Binary(.zip) | Encrypted file exchange |
| Product Guidance | Huawei iBMC (3.03.00.21) User Guide (Common Criteria Certification) | 3.03.00.21 | Document(.zip) | Encrypted file exchange |

Application note: the documentation *Huawei iBMC (3.03.00.21) User Guide (Common Criteria Certification)* contains preparative procedures and operational user guidance as sectors respectively.

1.4.2 Logical scope of the TOE

The iBMC FW supports a large range of security functionalities to facilitate various IT system integration, yet not all these functionalities are in the scope of the ST. As some of them are obsolete and may not be strong enough, or rarely used in most scenarios, they are remained to adapt some old IT systems which still do not migrate to state-to-art technologies, and they shall not be enabled and keep in disable in this certified configuration.

Following are distinction list of security functional specifications between the TOE and iBMC product.

| Security Functional Specification | | TOE | iBMC |
|-----------------------------------|-------------------------------|-----|------|
| TOE management | WebUI (Web user interface) | Y | Y |
| | CLI (command line) | Y | Y |
| | Redfish API | Y | Y |
| | IPMI | N | Y |
| | SNMP | N | Y |
| User authentication mechanism | password | Y | Y |
| | asymmetric cryptographic keys | Y | Y |
| | LDAP server | Y | Y |
| | Kerberos server | N | Y |

| | | | |
|--------------------|-------------------------------------|---|---|
| | 2 Factor Authentication (by Ukey) | N | Y |
| Security protocols | TLS 1.3/1.2 | Y | Y |
| | TLS 1.1/1.0 | N | Y |
| | SSH V2 | Y | Y |

The logical scope of the TOE including following security functionalities:

- ✓ Identification and Authentication
- ✓ Authorization
- ✓ Security Audit
- ✓ Cryptographic Support
- ✓ Security Management
- ✓ TOE access
- ✓ Trusted communication
- ✓ Secure updating based on RoT
- ✓ Secure booting based on RoT

1.4.3 Non-TOE hardware and software

All non-TOE hardware/software in operational environment including:

- ✓ A Huawei server device (hosts the TOE) equipped with iBMC, all supported Huawei server models are listed in Table 1-1.
- ✓ The iBMC hardware on which the TOE runs:
 - Hi1711, a dedicated iBMC SoC which integrates following modules:
 - Controller, providing the execution or processing environment for the FW as well as the management of the peripherals
 - Interface Controller, connecting to the platform controller hub, providing the local management network connectivity and provides the interfaces to users
 - eFuse, a one-time programmable IC component containing immutable credentials and implementing the trust anchor for booting and TOE FW update
 - ROM, an immutable memory containing the very initial firmware BootROM, this firmware is used to load and verify integrity of the first TOE FW package
 - RAM, is the fast volatile memory required for the runtime operation of the controller
 - BootROM program, a firmware stored in the ROM which use to load and verify the first FW package of the TOE. The BootROM can't be modified or updated during the whole life cycle of Hi1711 SoC, the version of the SoC 1711 also indicates the unique version of BootROM.

EFuse, ROM and credential data within them consist of root of trust (ROT), which used to support FW integrity verification, and their physical security is protected by operational environment

- EMMC Flash, a NVM (non-volatile memory) containing File systems including FW packages, log file and configuration data.
- ✓ One or several servers installed with syslog, NTP, LDAP, SFTP and FusionDirector services to act as syslog, NTP, LDAP, file and FusionDirector servers.
- ✓ A PC act as user client tools, it shall support web browser with TLS 1.2/1.3, SSH V2 client and HyperTerminal.

1.4.4 Summary of Security Features

1.4.4.1 Identification and Authentication

All TOE users – including the non-human user of FusionDirector server if present - are required to identify and authenticate to the TOE prior access can be granted according their role definition. The identification and authentication step shall always be conducted before a user can be authorized.

The TOE itself supports local authentication and external server based authentication invoking a LDAP.

For local authentication, the TOE supports mechanism both password and digital certificates based on SSH. Complexity of password and strength of asymmetric key are required.

For LDAP centralized authentication, the mechanism and security policy are configured and performed on the LDAP server, which is out of the scope of this ST.

1.4.4.2 Authorization

Whenever a TOE user has passed the authentication, the authorization assigns the appropriate role to the authenticated user. Each role shall have assigned TOE resources access and execution rights, respectively functional operation available. Only the functionality assigned to the authenticated role shall be authorized and enabled for execution.

1.4.4.3 Security Audit

The TOE generates audit logs of any security relevant event. The audit data contains sufficient information for tracing back to the origin or originator of the event and best its root cause of the event. Following events are recorded by the TOE:

- ✓ False authentications
- ✓ Data backup and restore
- ✓ Changes of the configuration

To each of the above events the following data shall be stored:

- ✓ Timestamp
- ✓ Event
- ✓ Result of the event
- ✓ User name (if possible)
- ✓ IP address (if possible)

The audit function is always active and cannot be disabled, audit records can be reviewed by only authorized and authenticated user, and can't be modified or deleted by any user. The TOE stores audit records in EMMC, and uploads them to a Syslog server according to its configuration.

1.4.4.4 Cryptographic Support

The TOE implements symmetric, asymmetric and hash-based cryptographic mechanisms for authentication, authorization, communication, as well as for confidentiality and integrity protection. The TOE is capable of generating the necessary keys (AES, RSA). For key generation, the TOE can use its own deterministic random number generator with an appropriate entropy quality.

1.4.4.5 Security Management

Security management includes aside from authentication, authorization and access control also the management of security related data, for example such as configuration profiles and runtime parameters.

The subsequent functions are defined as security management of the TOE:

- ✓ Management of user accounts, authentication data and role association
- ✓ Configuration of security related thresholds such as false authentication attempts
- ✓ Configuration of the authentication failure policy
- ✓ Customization of user roles and authorization, including the non-human user
- ✓ Configuration of authentication mechanisms and assigned trustworthiness services. For example servers for signature verification and similar
- ✓ Configuration of the cryptographic services and secured protocols
- ✓ Configuration of the log data and its backup
- ✓ Reading and copying rights for the audit log file
- ✓ Management and configuration of the session policy
- ✓ TOE configuration for the time synchronization with the NTP server

1.4.4.6 TOE Access

Each user session is started with the first authentication attempt which follows already a session policy. At delivery stage the session policy is preconfigured with a default setting, which can be modified by the administrator.

This policy includes:

- ✓ A configurable session timeout period leading to an automatic logout and disconnection of user.
- ✓ In case of abnormal behavior detection of the server, TOE users receive a warning information right at the point in time they access the login page. The warning information is configurable by authorized user.
- ✓ If the number of a user login-attempts exceed a configured threshold the corresponding account gets locked respectively disabled.

1.4.4.7 Trusted Communication

The TOE operates communication protecting protocols (HTTPS, TLS and SSH) with host external entities, including users, and the servers Syslog, LDAP and File. And, this comprises also all administration traffic to and from FusionDirector, the automated centralized server management.

It is essential that these protocols follow recommended standards to ensure correct operation of the following TOE functions:

- identification,
- authentication,
- authorization of user
- authorization of data,
- data integrity and
- data confidentiality.

A recommendation of appropriate protocols is given in [BSI-1] and [SOGIS-1].

1.4.4.8 Secure Booting Based on RoT

The TOE constitutes when correctly booted and having the TSF in place the trust anchor or root of trust to the host server. For that reason it is, right after powering-up or any other reset, absolute essential that the TOE conducts a booting procedure which ensures the correct operation of the TOE TSF, TOE operation and therewith the integrity of the host server.

This implements dedicated requirements for the booting procedure flow which are defined with dedicated SFRs in chapter 5.

1.4.4.9 Secure Updating Based on RoT

In order to keep the TOE up to date and maintain its protection against later discovered vulnerabilities, faults or simply for functional improvement, the TOE implements mechanisms for updating the TOE FW in a secure way.

The TOE shall ensure secure and trusted FW updating by verification of integrity and authentication of the updating package. The verification shall be performed based on a Root of Trust which is protected by an immutable memory hardware, and the updating shall not be executed if its verification and validation failed.

2 CC Conformance Claims

2.1 CC Conformance Claim

This ST is *CC Part 2* conformant [CC] with extended components defined in chapter 5, and *CC Part 3* conformant [CC]. The version of [CC] is 3.1R5.

The ST claims conformance to the EAL4 assurance package, augmented by ALC FLR.1.

No conformance to a Protection Profile is claimed.

3 TOE Security Problem Definition

3.1 Asset Definitions

The assets in the TOE that need to be protected are classified into the following types:

- ✓ TOE's FW integrity including its correct loading and TSF execution
- ✓ Any data stored, processed or generated
- ✓ TSF data
 - audit records
 - account information with administrator and operator credentials
 - ephemeral keys and random values used for the trusted communication
- ✓ Configuration data of the TOE including authorization
- ✓ Correct TSF protected operation

And following assets that need to be protected for TOE FW update mechanism:

- ✓ Remote data: Any data and/or code stored outside the TOE and deemed to be uploaded to the TOE.

3.2 Threats to the Assets Definition

The assets of this TOE type are threatened by following threat agents:

| Threat | Asset | Exploit result |
|--|--------------------------------|---|
| T.Unauthenticated An adverse user connected and authenticated successfully from outside to the server. | TSF data Configuration data | <ul style="list-style-type: none"> • The adverse user can access the TOE functions and is able to read or tamper or even destruct TOE or server configuration settings and/or audit logs. This comprises also attacks on the local management network interface, its servers and the connection to the central management server of the cloud or data centre. • The destruction of audit logs could make it even undetectable for the administrator that an unauthenticated access was conducted. |
| T.Unauthorized An adverse user authorized to the operator role achieves the rights of the administrator role. | TSF data Configuration data | <ul style="list-style-type: none"> • The administrator could in the worse completely abuse the TOE with administrator rights. Such scenario would comprise also attacks local management network interface, its servers and the connection to the central management server of the cloud or data centre. • This includes also adversely reconfiguration of the TOE; for example grant access to further attackers |

| | | |
|--|---------------------------------|--|
| | | and/or abuse the entire server for eavesdropping, tampering of TOE, host, common user data, construct DoS/DDoS, of other servers, or similar network based scenarios in a larger attack scenario. |
| <p>T.Intercept</p> <p>An adverse user able to intercept the communication between TOE, administrator, operator or servers on the local management network. This comprises also the machine-to-machine connection to the central management server of the cloud or data centre.</p> | TSF data | <ul style="list-style-type: none"> • The adverse user can reuse TSF data for replay or tampering of the before eavesdropped security relevant information interchanged between the entities, in order to gain access and the role of an authorized operator or administrator. • Tampered TSF data could lead to adversely reconfiguration of the TOE. This could include for example granting access to further attackers and/or abuse the entire server in a larger attack scenario. |
| <p>T.AccessData</p> <p>An adverse user able to access, read or tamper any data stored, communicated and operated by the TOE.</p> | Data | <ul style="list-style-type: none"> • Disclosed, deleted, or tampered data could lead to uncontrollable damage and loss of availability. • Eavesdropped credentials can lead to unauthenticated and unauthorized access and exploits of TOE functionalities. |
| <p>T.FaultyFW</p> <p>A wrong or damaged FW lead to vulnerabilities and unavailability of the TOE.</p> | Correct TSF-protected execution | <ul style="list-style-type: none"> • The induction of errors into the FW by other physical, e.g. voltage glitches, or nature, e.g. radiation events, can lead to loss of FW integrity, faulty behaviour, vulnerabilities and unavailability of service. The danger of fault induction by physical reasons is mainly given during initial FW loading, i.e. at booting time and when FW parts are loaded from the TOE's NVM on the bus. • Another threat appears if the FW would be loaded and executed without integrity check or version validation. Here, the loading from a memory module inside the TOE is in focus. A fault during conduct could lead to loading and execution of maliciously formed FW packages, where an attacker could exploit resulting vulnerabilities, take over control of the TOE in the worse, or achieve full stop of operation and services. <p>All of that can harm the TOE operation and could lead to full stop of operation and services.</p> |
| <p>T.Exploit</p> <p>An adverse user able to exploit possible known vulnerabilities of older FW version and may abuse the TOE based on these knowledge.</p> | Correct TSF-protected execution | <ul style="list-style-type: none"> • An attacker is able to identify operational dependencies or has knowledge about known vulnerabilities and can exploit them to circumvent one or more TSFs and abuse or exploit the TOE. • If vulnerabilities of older TOE FW versions are publicly known, the attacker could try to inject such vulnerable code in form of FW packages during start-up or download of adverse TOE FW updates. Thus he could exploit these vulnerabilities and abuse or exploit the TOE. |

The TOE FW update mechanism is threatened by following threat agent.

| Threat | Asset | Exploit result |
|---|-------------|---|
| <p>T.FaultyFWU</p> <p>A wrong or damaged FWU lead to vulnerabilities and unavailability of the TOE.</p> | Remote data | <ul style="list-style-type: none"> • This threat is generated in cases where the FW would be loaded and executed without integrity check or version validation. Loading refers here to a remotely conducted FW package update. An induced fault during conduct could lead to loading and execution of maliciously formed FW packages where an attacker could exploit resulting vulnerabilities, take over control of the TOE in the worse, or achieve full stop of operation and services. • This threat is generated when an attacker is able to observe and tamper the remote communication of the TOE FWU mechanism. This could lead to abuse of the update mechanism by <ul style="list-style-type: none"> ○ exchange or compromising of the FW package with a malicious code package, or ○ the attacker is able to manipulate the FWU mechanism in a way exceeding the desired FWU mechanism, so that other code is overwritten. <p>All of that can harm the TOE operation and could lead to full stop of operation and services.</p> |

3.3 Organizational Security Policies

Following policy is assumed to be effectively implemented or in place reducing the related threats to the assets:

3.3.1 P.ChangeForced

At the point in time the TOE is produced the TOE can be equipped with a default set of initial authentication credentials. These credentials are deemed to protect the TOE and therewith the server during delivery and storage.

For this reason the cryptographic quality of these credentials matters and in consequence, if the developer generates these default credentials outside the TOE, and imports them during the production phase, the developer should use appropriate equipment, for example a HW Security Module (HSM).

In addition, the developer is required to protect the default credentials from tampering and disclosure as long the legitimate user has not changed the default credentials into his individually defined credentials.

Either the TOE or alternatively the user guidance shall force respectively obligate the first administrator operating initially the TOE the first time to change the default into user individual credentials.

This provides a separation of responsibilities and preserves that even if developer's premises would have been compromised or an inside attacker disclosing initial credentials was discovered, the operated TOE remains authentication protected.

3.3.2 P.Ctrl_FWU

When the TOE implements an update functionality, it is required that at the point in time the trusted communication channel has been established, the authorized user shall control the usage of the FWU functionality. The control is required to protect stored, loaded or to be uploaded FW packages, patches and user data from disclosure, undesired deletion and manipulation.

Controlling the FWU functionality includes also the termination of the corresponding FWU session in a defined way - when the administration work has been finished - to preserve the communication channel from being abused.

3.4 Assumptions

The following assumptions are defined in order that the TOE can meet the targeted EAL:

3.4.1 A.PhysicalProtection

The TOE and its host as well as the server are protected against physical access from unauthorized users and attackers.

The servers Sys Log, NTP and LDAP or similar servers on the local management network are part of the equal physical protection means in the equally protected environment as used for the server hosting the TOE.

RMTs, PCs or other ICT devices operated by the authorized users and connected to the local management network run in a comparable physical protected environment.

3.4.2 A.ManagementNetwork

All instances on the local management network, especially the servers Sys Log, NTP and LDAP or similar, in the physical protected environment operate correctly and their services are available anytime.

Specifically, this assumption holds true for the local management network interfaces and the connection to the central management server of the cloud or data centre.

3.4.3 A.InternalConnections

The host internal connections on bus and network level between TOE and host, respectively between the management plane and the service plane, are available and work correctly.

3.4.4 A.NetworkSegregation

The local management network, carrying the management traffic of administrators, operators and connected servers, and the general data traffic of the common users are segregated within the physically protected environment.

The traffic segregation into different networks is immediately and automatically conducted as soon traffic from administrators or operators has been detected.

All traffic to and from remote administrators and operators via RMTs, PCs or other ICT devices is cryptographic protected.

3.4.5 A.TrustedPersonnel

It is assumed that personnel working as authorized administrators and operators shall be carefully selected for trustworthiness and trained for proper use of the TOE. These administrative users shall be technical competent, act thoroughly, do not act deliberately in a jeopardizing way, obey and maintain instructions provided by the TOE documentation.

As this TOE defines a management platform with which a server is operated and maintained, administrator users are assumed to connect and use the TOE in short time frames or even on a regular basis.

Following is assumed to properly maintain the TOE TSF:

- thorough maintenance procedures for the TOE are in place and executed, including
- the check for availability of security relevant SW, FW or patch updates and
- conduct such update either on regular basis or immediately in case of vulnerabilities as soon such update is provided.

In case of a given TOE FW update the original TOE configuration changes. This in turn requires the conduct of maintenance procedures for the belonging certificate. The authorized users are aware of this effect and the developer process follows the mandatory regulations depending on the chosen certification scheme.

3.4.6 A.HardwareROT

It is assumed that the hardware on which the TOE runs provides one-time-programmable (OTP) memories including ROM and eFuse to preserve the data of root of trust, so that the root of trust can be immutable from any attempting of modification. And data preserved in these memory modules can only be visited by the TOE,

It is assumed that BootROM program has already been installed correctly into the ROM, and the credential data used to verify the TOE has already been written correctly into the eFuse. Both BootROM and credential data along with OTP components host them consists of ROT for the TOE. The BootROM program shall verify the integrity and authenticity of the first booting FW package of the TOE before executing it.

Following are assumptions for FWU:

3.4.7 A.ResponsibleUser

The administrator user responsible for the conduct of the FWU mechanism operate appropriate measures to receive developer's notifications of updates. It is assumed that the administrator user supports the trusted communication channel between the TOE and the source of update.

3.4.8 A.ProtectedUpdate

The TOE implements update functionality, it is assumed that TOE FW update materials including files, code, data, documentation and also its provision on server side are treated in ways preventing from compromising, tampering and disclosure.

4 Security Objectives

4.1 Security objectives for the TOE

With respect to threats and assets, following objectives are required that the TOE meets the targeted EAL.

4.1.1 O.Authentication

The TOE requires identification and authentication of any user – including the automated, non-human management server of the cloud or data centre, connecting on the local management network - aiming to access the TOE functions. The authentication requirement shall be fulfilled for any operational status and the entire lifecycle.

If passing the authentication, the user shall get information about his role assignment.

The TOE shall support at least one authentication method which is recommended by [SOGIS-1] or [BSI-1].

This objective comprises also the non-human server authentication of the central management server of the cloud or data centre.

All user accounts and belonging authentication credentials shall be subject of the security management.

Application Note:

Identification means a user is identified and that the identified user belongs to the allowed user groups. Authentication means the assignment of the user to the defined roles administrator or operator. A common user has no access to the TOE. In many realization identification and authentication is done in one step which is acceptable.

End of the application note.

4.1.2 O.Authorization

After successful passing the authentication the user is assigned to a defined role.

Optionally, the authorization can require additional credentials for passing which shall be part of the security management. The additional authentication to the authorization step could for example be required when an administrator hierarchy has been implemented.

The authorization assigns to each role a dedicated predefined set of accessible and executable functions. Functions outside the assigned set shall not be accessible.

For this role definition, the administrator role has full access to all TOE functionalities. Deviations in form of establishing an administrator hierarchy are allowed.

This objective comprises also the authorization of the local management network connection to the central management server of the cloud or data centre – as a non-human user of the TOE.

4.1.3 O.FWIntegrity

Except the first booting FW is verify by the BootROM, the TOE shall execute only successful integrity verified FW parts following these requirements:

1. FW integrity verification shall be conducted at each powering up or reboot.
2. If remote download of FW updated is conducted, each downloaded package shall be verified for its integrity after the download has finished.
3. Even during the download on the protected channel, the individual transmitted data portions shall be integrity protected by the applied communication protocol.

There shall be no exception to skip such integrity verification.

4.1.4 O.FWValidation

The TOE shall validate the correctness of version of each FW package at the point in time its integrity has been successfully verified, but prior it is executed. This hold true for local loading from the NVM at each start-up and for FW package update using remote sources.

This means that the validation of each FW package prior execution shall have seen a two steps check:

1. Verification of the FW package integrity as objected above with O.FWIntegrity point 1, and
2. Validation of the FW package version control.

Thus this objective adds the version control on one hand and defines the fix sequence of the two tests.

4.1.5 O.Communication

The TOE shall operate communication protecting protocols between itself and remote administrators and operators. Deploying such protocol means that it establishes authenticated communication entities, authenticated data as well as integrity and confidentiality protection.

The TOE shall support or construct at least one secure protocol for which appropriate recommendations are endorsed by [SOGIS-1] or [BSI-1].

This objective for deploying a secured protocol comprises also the communication protocol of the local management network connection to the central management server of the cloud or data centre – as a non-human user of the TOE.

4.1.6 O.Audit

The TOE shall generate logs from security related events. Only the administrator is enabled to configure the type of stored audit data and the audit process including its backup.

The amount of logging data depend on the use case, configuration and capabilities of the Host. The definition of the minimum data to be logged is given in FAU_GEN.1 and applies for this TOE. The completeness and integrity of the audit data in the logging file are crucial for a possible later analysis of events.

Also the detail of data to be logged should in best case enable for a forensic analysis of the root cause of a security event. In any case the data should enable to identify the root cause of a security event. As a minimum set of data the TOE should fulfil [IEC-1, CR 2.8].

The log-file carrying the audit logs shall be access, integrity and confidentiality protected.

The only instance enabled to write audit logs shall be the TOE, even administrators can only read and copy the log-file.

For the case of a destructive security event, for example the server burned down, the log-file is required to be backed up outside the server in regular configurable intervals.

This objective ensures additionally that the log-file backup is conducted by the TOE in an integrity and confidentiality protected manner, so that only the enabled administrator is able to use the log-file backup in the intended way. This requires support from the operational environment.

4.1.7 O.SecurityManagement

The TOE provides authenticated and authorized security management respectively security configuration functions to the administrator. The minimum set of security management function shall include:

- User accounts with authentication data
- User authorization by assignment of functions to each user role
- User role notification and failure messages
- Limitation of authentication and authorization attempts and failure policy
- Definition of accepted certificate authorities for certificate and signature verification
- Configuration of cryptographic mechanisms used in secure protocols
- Configuration of the audit logs and audit process configuration in terms of memory control, backup
- Configuration of the session policy, for example termination of inactive logged in users
- Configuration of time synchronization, definition of NTP service
- User termination policy, destruction of secrets

Application Note:

All of the above listed security management functions shall apply also to the non-human TOE user of the central management server of the cloud or data centre

End of the application note.

4.1.8 O.LifeCycle

The TOE shall support following lifecycle aspects:

1. Set into operation:

At the point in time the TOE produced a set of default credentials protects the TOE from false identification, authentication and authorization. This set of default credentials protects the TOE during delivery, storage and during the installation phases, until the first legitimated administrator sets the TOE into operation and configures the TOE.

Since the default credentials can be generated by the developer environment the developer is in charge for their generation, storage and import. This is out of the scope of this TOE.

But, at the point in time of the first TOE configuration, the responsibility for the protected handling of the default credentials is removed from the developer and put to the legitimate administrator.

The TOE shall either force technically or obligate with the user guidance – as part of this TOE - the administrator to conduct the change from default to user defined credentials. This change marks the clear point in time where the legal responsibility for the credentials changes.

The administrator is from now on in responsibility for all possible harm caused by disclosed credentials.

2. Decommissioning:

At the point in time the administrator conducts decommissioning of the TOE, the TOE should either be physical securely destructed by the owner, or the Host, or the TOE provides functionality to securely wipe all user defined credentials and put it back in the original delivery condition with its default credentials. In the first case the user guidance should obligate the user and provide some guidance so that a decommissioned TOE cannot be reactivated and abused. For the second case the related functionality shall be part of the user guidance either of the Host or the TOE. The ST writer is required to clarify this.

4.1.9 O.Ctrl_FWU

The FWU mechanism shall only be conducted over a trusted communication channel with an authenticated and authorized user respectively communication entities. This can also include a trusted non-human user connected on the local management network. On top of the given trusted path when using the FWU mechanism, the FWU mechanism shall support data authentication, confidentiality and integrity protection of the transmitted user data as well as its validation and logging of the event.

4.2 Security Objectives for the Operational Environment

Following security objectives shall be met by the operational environment in order to assure the required protection for the correct operation of the TOE:

4.2.1 OE.ExclusiveProtHost

As the host respectively server's service plane usually conducts other services and interfaces with other components out of scope of the TOE, it is essential to define that no other management functionality or application shall be operated on the host's management plain, as long the TOE conducts its administration and management operation. A possible other server management instance could lead to interference and conflicts with the TOE and unpredictable behaviour of the service plane if management operations from different management instances are conducted at the same time.

This results in following objectives:

- As long the TOE conducts booting and any management operation on the host, no other application or SW shall by default be able to interact, to interfere, to inject other code, or conduct administrative or other management functions on the server's management plane respectively the TOE.
- The TOE host, the server's service plane and its operational environment provides physical protection of the management traffic on the local management network, as well as physical and logical segregation from common user traffic inside the server, i.e. separation of management from service plane.
- The booting integrity verification sequence shall be an offline, autonomous and inside-only procedure of the TOE without involving external services.

4.2.2 OE.CorrectHosts

The host respectively the server's service plane is out of control of the TOE and for this reason it is essential that the host provides its functionalities in the correct way which include the essential provisions of power, possibly clock frequencies and other essential physical interfaces.

During operation the host respectively the server's service plane does in no way jeopardize the correct operation of the TOE, meaning interfere the correct TOE operation.

The host respectively the server's service plane interacting with the TOE shall be available and work correctly as specified.

The host respectively the server's service plane provides isolation from any external connection for the time the TOE conducts the booting. This is given at each powering-up or after a reset of the host.

4.2.3 OE.PhysicalProtection

The TOE and its host as well as its local peripherals are protected against physical access from unauthorized users and attackers.

The servers Sys Log, NTP and LDAP or similar servers on the local management network are part of the equal physical protection means in the same environment as used for the server hosting the TOE.

As the central management server of the cloud or data centre operates also on the local management network, the equal physical protection means in the same, or comparable equal or even better protecting environment as used for the server shall be given.

The local and remote locations of the RMTs, PCs or other ICT devices used by administrators and operators are at least physically protected on a comparable level as the server location hosting the TOE.

4.2.4 OE.ManagementNetwork

All instances on the local management network, especially the servers Sys Log, NTP and LDAP or similar servers, in the physical protected environment operate correctly and their services are available for the TOE anytime requested.

This comprises also the availability of the non-human-initiated connection to the central management server of the cloud or data centre.

Aside of the availability of the non-human-initiated connection to the central management server of the cloud or data centre, it is essential that the operational policy of the central management server does not conduct any management operation conflicting with the management operation of the TOE.

Such seamless and non-conflicting operational policy is out of scope of the TOE and belongs to the administrator of the cloud or data centre.

4.2.5 OE.NetworkSegregation

The operational environment provides a segregation of traffic coming from remote administrators and operators on one hand and from other common user traffic on the other hand.

The operational environment provides a segregation of traffic coming from the non-human-initiated communication of the central management server of the cloud or data centre. However, this central administration traffic shall use the same local management network bearing the human administrator and operator traffic.

The segregation shall be implemented as soon the traffic's data type can be assigned. Administrator and operator user traffic are routed via the local management network to the TOE.

4.2.6 OE.TrustedPersonnel

Personnel working as authorized administrators or operators shall be carefully selected for trustworthiness and trained for correct use of the TOE. Administrators and operators shall be technical competent, act

thoroughly, do not interact deliberately in a jeopardizing way with the host and the TOE, and obey and maintain instructions provided by the TOE documentation.

Additionally, and if the TOE operates a FWU mechanism, administrator users have means in place to either receive developer notifications or check regularly check at developer provided sources or receive information in other ways upon developer provided TOE FWU packages.

The administrator user is capable to decide in responsible manner whether a FWU package needs to be installed and conducts the required action.

4.2.7 OE.CredentialProtection

It occurs at the point in time the TOE is produced that default, initial or other secret keys and other credential data are generated by the operational environment. If those are then imported into the host deemed for the use by the TOE' TSF, the following shall apply for the operational environment:

- Appropriate cryptographic methods respectively generation devices are applied and
- these data are either securely destroyed after import to the TOE or
- kept access controlled, integrity and confidentiality protected at all times the TOE is not finally decommissioned.

The equal objectives apply when the first administration user, respectively the device owner or, in other words, the customer overwrites the default credentials with customer individual credentials.

This preserves also the final legal domain separation between developer and user. From that point on the developer cannot be made reliable if the user generated credentials become disclosed, or are trivial values, or have characteristics enabling for being abused in other ways.

4.2.8 OE.AuditStorage

The TOE is required to fulfil the objective O.Audit and provides procedures for the logging of security events and the backup of the audit log-file.

The host and/or the operational environment shall preserve sufficient integrity protected memory space and the capability for unlimited backup of the log-files on the Sys Log server connected on the management network.

The treatment of log-files on the Sys Log server should be part of an organization policy which is outside the TOE. In any case the TOE shall be able to conduct the backup of log-files without limitation in size and intervals, or backup-blocking by unavailability by the environment.

The Sys Log server shall be available all time.

4.2.9 OE.FWU_Usage

For TOE updating, when accessing the TOE, the authorized user at his premises shall support the trusted communication channel. The channel is protected by means of authentication and confidentiality requiring interaction with the user.

4.2.10 OE.ImmutableROT

The SOC of iBMC shall provide immutable memories to hold root of trust data of the TOE. The root of trust consists of initial credentials used for the initial integrity verification, updating authentication. It therefore constitutes the most important part of the trust anchor for the TOE, the ROT root of trust.

Such immutable based HW module is required to form a trust anchor for the TOE and enable for secure booting and updating without usage of external services. The ROT contains of :

- certificates, public keys, other credentials and integrity verification reference values stored in eFuse which are used for verifying authentication and integrity of TOE booting and updating FW files.
- The very initial FW BootROM stored in ROM.

4.2.11 OE. IntegritybyBootROM

Each time power on or reboot, the BootROM program shall load the first FW package of the TOE, and verify integrity, authenticity and version validity of it. This FW package shall not be executed by BootROM if the verification is failed.

4.3 Overview: assumptions, policies and objectives for the environment

| Assumption, Threat or Organizational Policy | Objective for the Environment |
|---|--|
| A.PhysicalProtection | OE.PhysicalProtection |
| | OE.CorrectHosts |
| A.ManagementNetwork | OE.ManagementNetwork |
| A.InternalConnections | OE.ExclusiveProtHost |
| A.NetworkSegregation | OE.NetworkSegregation |
| A.TrustedPersonnel | OE.TrustedPersonnel |
| A.ResponsibleUser | OE.TrustedPersonnel |
| T.Unauthenticated | OE.CredentialProtection OE.AuditStorage |
| P.ChangeForced | OE.CredentialProtection |
| P.Ctrl_FWU | OE.FWU_Usage |
| A.ProtectedUpdate | OE.FWU_Usage |
| A.HardwareROT | OE.ImmutableROT |
| | OE.IntegritybyBootROM |

4.4 Threats and Security Objectives Rationale

The following table is deemed to put a rational for the presence of each objective covering the assigned threat.

4.4.1 Threats and Security Objective Rational for the base Package

| Threat | Objective | Rationale |
|-------------------|--|--|
| T.Unauthenticated | <p>O.Authentication</p> <p>O.Communication</p> <p>O.SecurityManagement</p> <p>O.Audit</p> <p>O.Lifecycle</p> | <p>O.Authentication</p> <p>The threat of uncontrolled users accessing the TOE functionality is countered by the requirement to implement at least one of the recommended authentication methods. The authentication is also in place for any operational status when a user tries accessing the TOE and this holds true for the entire lifecycle. Thus, this threat is countered anytime.</p> <p>O.Communication</p> <p>The application of one of the recommended and CB-endorsed protocols ensures that communication between TOE and a remote entity is only established after successful identification and authentication of each entity. By that it is not practical that attacker can capture such protected communication channel.</p> <p>O.SecurityManagement</p> <p>The security management defines the threshold of allowed failed access attempts and provides an appropriate reaction policy if the threshold is achieved or exceeded. This preserves that attackers cannot brute force or conduct unlimited analysis of authentication credentials.</p> <p>O.Audit</p> <p>Any login and failed login attempt are documented in the log. This enables administrators to identify misbehaviour of users and tracing back attack attempts to the root cause or individual account. Such audit logs enable administrators to take appropriate action and defence against adverse users.</p> <p>O.LifeCycle</p> <p>The TOE shall protect itself from being tampered or modified even during delivery, storage and installation phases of its hosts. For that reason, the TOE provides initial protection with a secure deemed default configuration including set of default credentials requiring identification, authentication and authorization even at the first time the TOE is set into operation.</p> <p>Even after decommissioning the TOE shall not provide the condition for being reactivated and abused. This is prevented with functionality to wipe all user credentials securely and put it into the original delivery conditions with respect to default credentials and default configuration.</p> |

| Threat | Objective | Rationale |
|----------------|--|---|
| T.Unauthorized | O.Authorization O.SecurityManagement O.Audit | <p>O.Authorization</p> <p>Unauthorized execution of functionalities is rendered not practical as each authenticated user has assigned a dedicated role which is equipped with defined set of functionalities accessible and executable. Functions outside the defined set are not accessible as those belong to a different user role. Thus, only authorized functions can be executed.</p> <p>O.SecurityManagement</p> <p>Administrators define and configure with the security management the allowed functionality for each user role. As an option the security management can require another authentication step prior the set of functions is made accessible. The presence of a dedicated security management protects against adverse role extension or privilege level elevation, and prevents from the use of not allowed functions for an identified role.</p> <p>O.Audit</p> <p>Administrative actions are logged with sufficient detailed information and enable by that to trace back to malicious events and identify adverse users. The audit process shall ensure that even no administrator can tamper or delete the log file.</p> |
| T.Intercept | O.Communication O.SecurityManagement | <p>O.Communication</p> <p>The threat of interception is countered by using recommended and CB-endorsed protocols. Such a protocol preserves that it is not practical for an attacker to exploit identification and authentication of the entities establishing a secured communication, even the attacker eavesdrops or tampers the communication.</p> <p>O.SecurityManagement</p> <p>The administrator configures with the security management the communication with remote users that only the appropriate secure protocols with predefined entities having the correct credentials can be established. The security management prevents the establishment of communication channels using inappropriate protocols for administrator or operator management tasks.</p> |
| T.AccessData | O.Communication O.SecurityManagement | <p>O.Communication</p> <p>The use of a recommended and CB-endorsed protocol preserves that data transferred via this protocol is authenticated as well as integrity and confidentiality protected. Even deletion of data during transfer can be detected. As data content of administrator's or operator's management traffic can contain user and personal data, the protection of those from disclosure is crucial in technical sense but also for following the GDPR.</p> |

| Threat | Objective | Rationale |
|------------|--|--|
| | | <p>O.SecurityManagement</p> <p>The administrator configures with the security management the communication with remote users. This configuration comprises that only the appropriate recommended and CB-endorsed protocols with integrity and confidentiality protection are applied. The security management prevents the establishment of communication channels using vulnerable and cryptographic weak protocols for administrator or operator management tasks.</p> |
| T.FaultyFW | <p>O.FWIntegrity</p> <p>O.FWValidation</p> <p>OE.ImmutableROT</p> <p>OE.IntegritybyBootROM</p> | <p>OE.ImmutableROT</p> <p>The ROT data is protected from modification by immutable hardware memory components, this data is used to verify the integrity and authenticity of the TOE.</p> <p>O.FWIntegrity</p> <p>The integrity of each FW package shall be verified for its integrity after loading but before execution. This prevents from undefined TOE behaviour by the undetected induction of faults during the booting sequence and remote update.</p> <p>Integrity verification before execution of a loaded TOE FW package shall be applied in any case and covers also the FW packages remotely downloaded for patching and SW updates.</p> <p>OE.IntegritybyBootROM</p> <p>The integrity of the first FW package during booting shall be verified after loading by BootROM program before execution.</p> <p>O.FWValidation</p> <p>This objective introduces the fix sequence of first integrity verification and then version validation to prevent the execution of integrity verified FW packages with wrong or obsolete version information. This prevents the exploitation of possibly known vulnerabilities of older versions coming along with a correct integrity verification.</p> |
| T.Exploit | O.FWValidation | <p>O.FWValidation</p> <p>This objective introduces the fix sequence of first integrity verification and then version validation to prevent the execution of integrity verified FW packages with wrong or obsolete version information. This prevents the exploitation of possibly known vulnerabilities of older versions coming along with a correct integrity verification.</p> <p>And an attacker will fail to induce disabled FW version packages during start-up and updates.</p> |

4.4.2 Threats and Security Objective Rational for the FWU package

| Threat | Objective | Rationale |
|-------------|------------|--|
| T.FaultyFWU | O.Ctrl_FWU | <p>O.Ctrl_FWU</p> <p>This objective ensures that on hand the remotely transported FW package stem from an authenticated and authorized source of the administrator user, and on the other hand, that the transmitted content is neither compromised, tampered or disclosed.</p> <p>This objective includes also that the desired and integrity verified FW package is installed by validating the right version. Also, it is ensured, that the communication path used, is protected from being compromised, tampered or disclosed, and last, that the overall update process is logged correctly.</p> |

4.5 Assumption, Polices and Objective for the Environment Rationale

The following table is deemed to put a rational for the presence of each objective for the environment covering the assigned threat.

4.5.1 A, P and OE Rationale for the base package

| Assumption, Threat or Organizational Policy | Objective for the Environment | Rationale |
|---|---|--|
| A.PhysicalProtection | <p>OE.PhysicalProtection</p> <p>OE.CorrectHosts</p> | <p>Since the TOE is not enabled to implement physical protection means against invasive or non-invasive attacks at local premises, the host is required to implement means of the OE.PhysicalProtection to cover this assumption.</p> <p>For the equal reasons, meaning the TOE has not implement checking means to verify the correctness of the host when operating other applications or even the host HW, the OE.CorrectHosts must fulfil the assumption A.PhysicalProtection.</p> |
| A.ManagementNetwork | OE.ManagementNetwork | <p>The justification of A.ManagementNetwork is that the TOE acts as a distant client and cannot verify the correctness of operation of the connected servers on the management network. However, dedicated security functionality with respect to audit, time synchronization, time stamps and authentication relies on the correct operation of the servers on the management network.</p> <p>For that reason, the means of OE.ManagementNetwork shall cover the aspects of the assumption A.ManagementNetwork.</p> |

| Assumption, Threat or Organizational Policy | Objective for the Environment | Rationale |
|---|--|---|
| A.InternalConnections | OE.ExclusiveProt Host | <p>As the TOE is operated in a host operating other applications, an interfering, eavesdropping or abuse of internal communication inside the host cannot be detected by the TOE. For that reason, the assumption is justified that the host is exclusively operated for the TOE only as long the booting takes place until the TOE has achieved its correct full functionality, and during the times the TOE conducts the desired management operation on the host.</p> <p>And, it is also required that the host provides segregation from other user traffic and protection of the communication between TOE and host as long the TOE executes its management functions for the host. Both is covered with OE.ExclusiveProtHost and the assumption A.InternalConnections is met by this.</p> |
| A.NetworkSegregation | OE.NetworkSegregation | <p>As soon administrator or operator management traffic enters the operational environment the traffic is to be separated and protected from other user traffic in order to avoid adverse redirection, analysis of meta-data, DoS/DDoS or similar. For that reason, this kind of traffic is to be protected and separated by the means given in OE.NetworkSegregation. These means meet the assumption A.NetworkSegregation.</p> |
| A.TrustedPersonnel | OE.TrustedPersonnel | <p>Correct configuration and operation of the TOE is crucial for the trustworthy operation of the host and therewith to the entire general administrated server. For that reason, the users conducting the management functions shall be able to deploy the operations correctly and the users shall not implement vulnerabilities by deliberate or erroneous configurations. The OE.TrustedPersonnel meets the assumption A.TrustedPersonnel.</p> |
| A.HardwareROT | OE.ImmutableROT OE.IntegritybyBootROM | <p>The integrity of each FW package shall be verified with ROT after loading but before execution. Therefore immutability is a key attribute of ROT to ensure integrity verification. The OTP memories integrated in the iBMC hardware which is used to preserve ROT shall prevent any attempting to modify this data. And the immutable BootROM program shall verify the integrity and authenticity of the first FW package of TOE before executing it.</p> <p>The OE.ImmutableROT and OE.IntegritybyBootROM meets the assumption A.HardwareOTP.</p> |
| T.Unauthenticated | OE.CredentialProtection | <p>At the point in time the TOE is produced, the cryptographic means for authentication and authorization are equipped with default credentials. Disclosure of these credentials from developer's premises or just using trivial values could</p> |

| Assumption, Threat or Organizational Policy | Objective for the Environment | Rationale |
|---|-------------------------------|---|
| | OE.AuditStorage | <p>compromise the TOE and in consequence the entire general administrated server, as an adverse user could access then the administrator role. For this reason, the cryptographic quality of these credentials matters. Thus OE.CredentialProtection contributes in the defence against T.Unauthenticated.</p> <p>In the event of unauthenticated access, for example with tampering, adverse modification or even destruction of the host and therewith destruction also of the host-stored log-file, it would be essential to exploit the backup of log-file. For that reason, the operational environment provides always sufficient and integrity protected memory space to conduct these backups. Thus, the environmental objective OE.AuditStorage and the OE.CredentialProtection, both together contribute to the protection against T.Unauthenticated.</p> |
| P.ChangeForced | OE.CredentialProtection | <p>At the point in time the TOE is produced, the cryptographic means for authentication and authorization could be equipped with default credentials. These credentials are deemed to protect the TOE during storage and delivery. For this reason, the cryptographic quality of these credentials matters.</p> <p>In addition, the legitimate administrator user shall ensure that also no remotely connected adverse user from developer's personnel having knowledge of and abusing the default credentials can compromise the general administrated server. The administrator user shall be able to import self-created credentials to replace the developer default credentials. In all cases, the cryptographic quality of these credentials matters.</p> <p>OE.CredentialProtection takes care of these requirements and fulfils P.ChangeForced.</p> |

4.5.2 A, P and OE Rationale for the FWU package

| Assumption, Threat or Organizational Policy | Objective for the Environment | Rationale |
|---|-------------------------------|---|
| P.Ctrl_FWU A.ProtectedUpdate | OE.FWU_Usage | <p>The TOE provides remote FW update functionality initiated and conducted by an authorised, but distant user. Distant means that the user environment, out of which the update is conducted, is out of control of the TOE. Thus, a working place can be operated by different users, the authenticated and authorised, and the not-authenticated and not-authorised.</p> <p>For that reason, the user environment is required to support the identification, authentication and authorisation means for the trusted communication channel.</p> |

| Assumption, Threat or Organizational Policy | Objective for the Environment | Rationale |
|--|--------------------------------------|---|
| A.ResponsibleUser | OE.TrustedPersonnel | The administrator user shall take care for FWU packages provided by the developer in order to keep the TOE up-to-date with the most recent FWU packages. The operational environment provides the required information and the means to receive TOE FWU packages by the responsible administrator user. The administrator user is also capable to decide in responsible manner upon the conduct of the FWU. The OE.TrustedPersonnel meets the assumption A.ResponsibleUser. |

5 New and Extended Components Definition

These new and extended components are defined in this ST, they are collected here for easy readability.

5.1 Extended Functional Components Definition

5.1.1 FPT_ARB Anti-Rollback

The [CCP2] contains a SFR (FDP_ROL) which enables to roll back to a previous version which is the opposite of what this new defined family requires to achieve. In certain cases a rollback of SW to an older version may be a use case, but this does not hold true for this TOE.

For that reason this family FPT_ARB is newly defined as follows: The TOE shall implement a validation procedure that only the desired, correct and up-to-date FW is executed. This protects from execution of possible vulnerable, obsolete FW package versions, which have been checked for integrity correctness but do not provide up-to-date protection.

The SFR family FPT_ARB requires the presence of a version control procedure prior each FW package comes to execution.

| | |
|--------------------------------|--|
| Family behavior: | This family ensures that only at runtime validated FW package versions are executed. |
| Component leveling: | <div style="display: flex; align-items: center; gap: 10px;"> <div style="border: 1px solid black; padding: 5px; text-align: center;">FPT_ARB Anti-Rollback</div> <div style="font-size: 20px;">→</div> <div style="border: 1px solid black; padding: 5px; text-align: center;">1</div> </div> <p>This family consists of one component only.</p> |
| Management: | There are no management activities foreseen. |
| Audit: | Minimal: Attempts to rollback TOE FW |
| FPT_ARB.1 Anti-Rollback | |
| Hierarchical to: | No other components. |
| Dependencies: | FAU_GEN.1 |
| FPT_ARB.1.1 | The TOE shall implement a procedure to verify the version of each FW package after its integrity has been successfully verified. This holds also for TOE FW update packages stored by the loader mechanism. If the version is verified the FW package is validated. |
| FPT_ARB.1.2 | Each FW package version shall be extracted and verified with its <i>[selection: reference value, computed version value, administrator updated version value]</i> stored in the <i>[selection: OTP memory, ROM, NVM]</i> . An exception from this is the very initial FW package which shall be read from the ROM only. |
| FPT_ARB.1.3 | A FW package version can have an equal or higher number as the reference number, but never a smaller number. If the version number is smaller the TOE shall stop the loading sequence and achieve a secure state. |

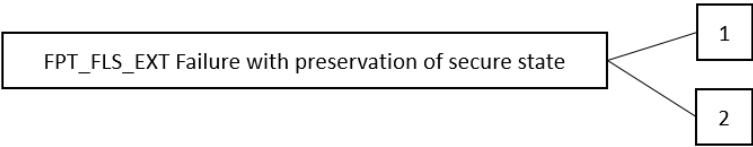
Application Note:

Executing only the correctly validated FW version is crucial for the correctness of the TSF. For that reason and if the version number is smaller than the reference number the TOE should proceed the secure state. This means for example that a smaller version number should be treated as if a FW package integrity error would have occurred. In this case the TOE shall follow then the requirements as defined in FPT_FLS_EXT.1.

End of note.

5.1.2 Extended Family FPT_FLS Failure with preservation of secure state

The extended component defined here is an extension of the SFR family FPT_FLS in [CCP2] and consists to two components, whereas the first component applies to TOE booting and the second one applies to TOE updating.

| | |
|---------------------|--|
| Family behavior: | Components in this family address the requirements for the TSF for preservation of a secure state in the face of identified errors. |
| Component leveling: |  <p>This family consists of two components, of which the first requires TSF preserve a secure state in face of booting error, and the second of updating error.</p> |
| Management: | There are no management activities foreseen. |
| Audit: | <p>The following minimum actions shall be auditable as FAU_GEN Security audit data generation is claimed:</p> <ul style="list-style-type: none"> • Logging of the type of identified error • If applicable, which part of the TOE FW had caused the detected error |

5.1.2.1 FPT_FLS_EXT.1 Failure with preservation of secure state - booting

This SFR applies in any case regardless whether [ISO TR] or [ISCI WG1] was chosen.

| | |
|--|--|
| FPT_FLS_EXT.1 Failure with preservation of secure state - booting | |
| Hierarchical to: | No other components. |
| Dependencies: | No dependencies |
| FPT_FLS_EXT1.1 | <p>The TSF shall preserve a secure state when the following types of failures occur: <u>[FW package integrity and FW version validation, detected after TOE FW loading, non-availability of the cryptographic coprocessor or other shortcomings]</u>.</p> |

| | |
|--|--|
| | <p><u>The TOE shall run following decision-sequence to achieve either the secure state or normal operation condition:</u></p> <ol style="list-style-type: none"> 1. <u>Discard the FW part causing the integrity error.</u> 2. <u>Conduct a security event logging.</u> 3. <u>Conduct reloading of this FW-part from the memory source-2 storing a fallback respectively reserve image of the FW</u> <ol style="list-style-type: none"> a. <u>Every restart from source-2 shall trigger according logging of the integrity fault.</u> b. <u>After logging restart the TOE from power-off condition or continue with 5.</u> 4. <u>In case of successful integrity verification:</u> <ol style="list-style-type: none"> a. <u>Logging of the successful verification</u> b. <u>Achieve normal operation condition</u> c. <u>Optional: Enable the administrator to conduct specific self-testing to conduct root cause analysis</u> 5. <u>Stop operation of the TOE and shutdown all services of the TOE to achieve a secure state, where only a local administrator is enabled to restart the TOE in a maintenance mode.</u> |
|--|--|

Application Note:

The ST writer shall specify the applied integrity verification procedure. It is crucial that the FW integrity is verified after loading of each part, but even before execution. In case of a detected integrity error the TOE shall achieve a secure state and never execute the faulty FW part. According to claim [CCP2], the term “secure state” refers to a state in which the TSF data are consistent and the TSF continues correct enforcement of the SFRs. A faulty executed FW part cannot ensure correct TSF enforcement.

End of the application note.

5.1.2.2 FPT_FLS_EXT.2 Failure with preservation of secure state - update

| | |
|---|---|
| FPT_FLS_EXT.2 Failure with preservation of secure state - update | |
| Hierarchical to: | No other components. |
| Dependencies: | No dependencies |
| FPT_FLS_EXT2.1 | <p>The TSF shall preserve a secure state if the conduct of verification and validation of the new downloaded package fails or any other error occurs during the conduct of the update procedure.</p> <p>The TOE shall run following sequence to achieve either the secure state or normal</p> |

| | |
|--|--|
| | <p>operation condition which is define also as secure state:</p> <ol style="list-style-type: none"> 1. Trigger a TSF initiated termination of the communication channel to the update server. 2. Conduct a security event logging. 3. Discard the downloaded FW package causing the integrity error. 4. Notify the administrator user (human/non-human) from the fail and that the downloaded FW package with version information was discarded. 5. Close the FWU functionality and resume normal operation state. 6. Conduct a security event logging. 7. For the case normal operation state can no more be achieved, stop operation of the TOE and shutdown all services of the TOE to achieve a secure state, where only a local administrator is enabled to restart the TOE in a maintenance mode. <ol style="list-style-type: none"> a. The administrator should follow a procedure in the user guidance to examine whether there is a persistent failure and b. should have the possibility to decide for or against a repair operation. <p style="margin-left: 40px;">And, if deciding against repair, the decommissioning case applies.</p> |
|--|--|

5.1.3 FPT_ROTSEB Root of trust secure booting

The following new defined SFRs provide the functional requirements, explanations of secure booting based on root of trust.

The secure booting procedure relies on the root of trust which shall be immutable. The immutability attribute is supported by OTP HW components.

And, these requirements are of static nature and constitute the first package of pre-conditional requirements, FPT_ROTSEB.1. The second component of this family describes the requirements and characteristics of the procedure respectively dynamic flow of secure booting and constitute the second element of this family, FPT_ROTSEB2.

| | |
|---------------------|--|
| Family behavior: | This family defines the requirements for the conduct of booting the TOE so that as a result the TSF operate correctly and the desired TOE services can be provided in the protected way. |
| Component leveling: | <div style="display: flex; align-items: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">FPT_ROTSEB Root of trust secure booting</div> </div> <p>This family consists of two components, of which the first provides the preconditions and functional limitations or characteristics, and the second the atomic sequence of conduct.</p> |
| Management: | There are no management activities foreseen. |
| Audit: | There are no actions defined to be auditable. |

5.1.3.1 FPT_ROTsb.1 Root of trust secure booting procedure

This SFR provides ROT based prerequisites and other static set-ups and limitation which shall be in place as fundament for the secure booting procedure.

| FPT_ROTsb.1 Root of trust secure booting procedure | |
|---|---|
| Hierarchical to: | No other components. |
| Dependencies: | FPT_FLS_EXT.1 failure with preservation of secure state extended, FRU_FLT.1 degraded fault tolerance, FPT_RCV.2 Automated recovery |
| FPT_ROTsb.1.1 | <p>All subsequent TOE FW packages except of the first one, shall be verified for their integrity before execution. The integrity verification shall be conducted by digital signature verification. This requires that each single FW package comes with an according digital signature respectively certificate. The following shall apply:</p> <p>[selection:</p> <p>using reference data stored in the root of trustor</p> <p>using trusted data which shall be hard-coded in previously integrity verified and executed FW-package.</p> <p>]</p> |
| FPT_ROTsb.1.2 | <p>The TSF shall verify the digital signature of any immutable FW code using digital signature algorithm specified in FCS_COP.1/ROTSig, The signature verification implicitly requires the according hash calculation of the code in question. The procedures shall use the hash algorithm as specified in FCS_COP.1/PubkeyHash.</p> |
| FPT_ROTsb.1.3 | <p>The TSF shall verify each certificate or public key or hash value of a public key hard coded in each FW package loaded from NVM by</p> <p>[selection:</p> <p>comparing it with the public key within the ROT or the previous executed FW package;</p> <p>calculating its hash value with hash algorithm specified in FCS_COP.1/PubkeyHash, then comparing the hash value with the public key hash within the ROT or the previous executed FW package;</p> <p>a combination of both;]</p> <p>before the signature of the FW package is verified with these data.</p> |
| FPT_ROTsb.1.4 | <ol style="list-style-type: none"> 1. During the entire procedure of secure booting the TOE does not interact with any other module or device of the Huawei server' service plane. 2. The secure booting is an offline process and shall not rely on other services. |
| FPT_ROTsb.1.5 | <ol style="list-style-type: none"> 1. For the case of failure and following the requirements of FPT_FLS_EXT.1, the TOE FW shall provide one fallback respectively reserve image of the TOE FW. Note: The reserve image can only be used if the essential parts for TOE operation have been successfully executed and the fault has occurred at a later stage of the booting sequence. End of note. 2. This reserve image shall be accessible for the TOE FW only during the booting phase and not during normal operation. 3. However, in case of TOE FW update it should be able to be update also the fallback image by the authorized administrator. 4. It shall be ensured that booting from this fallback or reserve image is only |

| | |
|----------------|---|
| | possible during the conduct of FPT_FLS_EXT.1. |
| FPT_ROTSTB.1.6 | The TOE shall provide a maintenance mode in case the requirements of FPT_FLS_EXT.1 are followed in fail state, and enable a locally present human administrator to enter this maintenance mode. |

5.1.3.2 FPT_ROTSTB.2 Root of trust secure booting atomic sequence

It is essential for the TOE integrity that the sequence required by the following SFR is conducted in atomic manner. The reason for this requirement is that the TOE cannot and shall not rely on any TOE-external service at this early stage. Any interference from the host or even external entities implement a potential vulnerability and point of attack.

Atomic in this sense means that the TOE booting sequence has no functional or interrupt based entry point and interface, and is able to detect failure during the conduct of its sequence and lead to the defined and required TOE failure reaction.

The sequence loads one FW package after the other until the current FW package does no longer indicate or include the public certificate or public key for the next following FW package, or another condition or flag terminates the chain of trusted loading.

| FPT_ROTSTB.2 Root of trust secure booting atomic sequence | |
|--|--|
| Hierarchical to: | No other components. |
| Dependencies: | FPT_FLS_EXT.1 Failure with preservation of secure state extended, FRU_FLT.1 Degraded fault tolerance, FPT_RCV.2 Automated recovery |
| FPT_ROTSTB.2.1 | <ol style="list-style-type: none"> The TOE shall load one FW package after the other and verify its integrity with validation of the belonging digital signature or public key. When the integrity of one package has been verified it is essential that the version number of this package is validated before the next package is loaded. <p>In case of identified failure following shall apply: If the verification of integrity of one package or the check of the version number fails, the SFR FPT_FLS_EXT.1 applies and the booting sequences is terminated therewith.</p> |

Note:

If the secure booting fails there is also no functionality to conduct logging of the security event.

End of note.


Application Note for the SFR FPT_ROTSTB.2.2:

An unbreakable loop of restarts can be interpreted as secure state, as none of the TOE functions is available outside the Host, as both still are not in full operation. In consequence none of the functionality can be addressed or requested from outside the host and thus an exploit of this situation is not practical. This includes also the logging of security events which cannot be conducted at this early stage.

End of the application note.

5.1.4 FPT_ROTSTU Root of trust secure updating

The following new defined SFRs provide the secure updating functional requirements, explanations and provide a reference for the ST writer without prescribing the concrete design.

| | |
|---------------------|--|
| Family behavior: | This family defines the requirements for the conduct of updating the TOE FW so that as a result the TSF operate correctly and the desired TOE services can be provided in the protected way. |
| Component leveling: |  <p>This family consists of two components, of which the first provides the preconditions and functional limitations or characteristics, and the second the atomic sequence of conduct..</p> |
| Management: | <p>FPT_ROTUSU.1</p> <p>The following actions could be considered for the management functions in FMT:</p> <p>a) Ability to update the TOE</p> |
| Audit: | There are no actions defined to be auditable. |

5.1.4.1 FPT_ROTUSU.1 Root of trust secure updating procedure

This SFR provides the HW based prerequisites and other static set-ups and limitation which shall be in place as fundament for the secure booting procedure.

| FPT_ROTUSU.1 Root of trust secure updating procedure | |
|---|--|
| Hierarchical to: | No other components. |
| Dependencies: | <p>FPT_FLS_EXT.1 failure with preservation of secure state extended,</p> <p>FRU_FLT.1 degraded fault tolerance,</p> <p>FPT_RCV.2 Automated recovery</p> <p>FCS_COP.1 Cryptographic operation</p> <p>FPT_TST_EXT.1 Basic TSF Self Testing</p> |
| FPT_ROTUSU.1.1 | The TSF shall apply data authentication and verify the integrity of a FW update package before writing it to the file system. The integrity verification shall be done by means of a digital signature. The verification reference data shall be taken from the root of trust (ROT). |
| FPT_ROTUSU.1.2 | <p>The TSF shall verify the public key within the FW updating package by <i>[selection:</i></p> <ul style="list-style-type: none"> • <i>comparing it with the public key within the ROT or the previous executed FW package;</i> • <i>calculating its hash value with hash algorithm specified in FCS_COP.1/PubkeyHash, then comparing the hash value with the public key hash within the ROT or the previous executed FW package;</i> • <i>a combination of both;]</i> |

| | |
|----------------|---|
| | before the signature of the FW package is verified with these data. |
| FPT_ROTUSU.1.3 | The TSF shall verify the digital signature of the FW updating package using digital signature algorithm specified in FCS_COP.1/ROTSig, The TSF shall validate the version of the newly loaded TOE FW package. The interim storage of the new TOE FW package required during verification and validation procedures shall not overwriting the original TOE FW image. |
| FPT_ROTUSU.1.4 | For the case of failure and following the requirements of FPT_FLS_EXT.2, the TOE FW shall stop the updating and stay with the original version. |
| FPT_ROTUSU.1.5 | 1. Only when the updated TOE FW has been successfully validated and a first reboot with this updated package was successful, the former FW image shall be replaced with updated TOE FW. 2. To ensure all TSF work correctly with the new FW package the TOE shall conduct self-testing as defined with FPT TST EXT.1.2. |
| FPT_ROTUSU.1.6 | The TOE shall provide a maintenance mode in case the requirements of FPT_FLS_EXT.2 are followed in fail state, and enable a locally present human administrator to enter this maintenance mode. |

5.1.4.2 FPT_ROTUSU.2 Root of trust secured updating - conduct

This SFR provides requirements for the secured conduct of the TOE FW Updated mechanism.

| FPT_ROTUSU.2 Root of trust atomic secure updating-conduct | |
|--|--|
| Hierarchical to: | No other components. |
| Dependencies: | None |
| FPT_ROTUSU.2.1 | 1. All cryptographic operations conducted during the updating phase respectively the FWU operation, shall be operated on the TOE. 2. There shall be no other TOE external component in use by the FWU mechanism. 3. This means on one hand that no other host component shall be able to operate and use the TOE cryptographic coprocessor during the TOE updating sequence, and, on the other hand, that the TOE shall not make use of another non-TOE cryptographic coprocessor located on the host. |
| FPT_ROTUSU.2.2 | 1. During the entire procedure of the FWU operation and updating, the TOE does not interact with any other module or device of the server and service plane. 2. The FWU operation and updating shall not rely on other services such as the host HW services for random numbers, voltage, clock, and similar provisions. None of these services shall interfere on FW functional or data level. |

5.1.5 Extended FPT_TST_EXT.1 TSF testing

The extended component defined here is a simplified version of FPT_TST.1 in [CCP2] as defined by [CEN-1] but extended with one important element to meet the patching requirements.

| | |
|------------------|---|
| Family behavior: | Components in this family address the requirements for self-testing the TSF for selected correct operation. |
|------------------|---|

| | |
|-------------|---|
| Management: | There are no management activities foreseen. |
| Audit: | The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST: <ul style="list-style-type: none"> • Indication that TSF self-test was completed. |

| FPT_TST_EXT.1 Basic TSF Self Testing | |
|---|--|
| Hierarchical to: | No other components. |
| Dependencies: | No dependencies |
| FPT_TST_EXT1.1 | <p>The TSF shall run a suite of the following self-tests <i>[selection:</i> <i>during initial start-up (on power on),</i> <i>periodic during normal operation,</i> <i>at the request of the authorized user, at the conditions</i> [assignment: conditions under which self-tests should occur]<i>]</i> to demonstrate the correct operation of the TSF.</p> <p><u>These self-testing shall comprise:</u></p> <ol style="list-style-type: none"> 1. <u>I</u> 2. <u>TOE FW integrity testing,</u> 3. <u>[assignment: list of additional self-tests run by the TSF].</u> |

5.2 Extended Assurance Components Definition

None.

6 Security Requirements for the TOE

6.1 Conventions

The following conventions are used for the completion of operations:

- ~~Strikethrough~~ indicates text removed as a refinement
- (underlined text in parentheses) indicates additional text provided as a refinement.
- If not being the headline of the SFR itself, **Bold text** indicates the completion of an assignment.
- *Italicised and bold text* indicates the completion of a selection.
- Iteration/Identifier indicates an element of the iteration, where Identifier distinguishes the different iterations.
- Normal text applies unchanged from the SFR definition in [CCP2] and extended components defined in chapter 4.

6.2 Security Functional Requirements

6.2.1 Security Audit (FAU)

6.2.1.1 FAU_GEN.1 Audit Data Generation

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

- ~~a) Start up and shut down of the audit functions~~
- b) All auditable events for the *not specified* level of audit; and
- c) **The following auditable events:**
 1. **User activities**
 - a. **Login and logout**
 - b. **Addition, deletion and modification of users**
 - c. **User credentials change**
 - d. **User role change**
 - e. **User disable and enable**
 2. **System Management Events**
 - a. **Protocol of the system management functions as defined in FMT_SMF.1 Specification of Management**
 - b. **Power on, power off and restart of the TOE**

Application Note: With respect to FAU_GEN.1.1: The start-up and shut down of audit functions does not apply as the audit functionality shall be active immediately after start-up and remains active all times. This means that even the administrator shall not have a possibility to stop or shut down the audit functionality.

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, **Operation Type (if applicable), Operation Object (if applicable), Access IP Address (if applicable), User Name (if applicable).**

Application Note: With respect to FAU_GEN.1.2: The parenthesis 'if applicable' shall be understood as 'whenever an event can be associated with the specified information'. For example, if an event can be associated with a User Name, then the event shall be recorded and the audit information shall contain the User Name. If the event cannot be associated with the User Name, the event shall be recorded and the audit information shall not contain User Name information. If multiple conditional information can be associated with an event (e.g. interface and User Name can be associated with an event), all conditional information shall be contained in the audit information logging the event.

6.2.1.2 FAU_GEN.2 User Identity Association

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

FAU_SAR.1.1 The TSF shall provide **Administrators** with the capability to read **all information** 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

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_STG.1 Protected Audit Trail Storage

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

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

6.2.1.6 FAU_STG.3 Action in Case of Possible Audit Data Loss

FAU_STG.3.1 The TSF shall **overwrite the oldest records** if the audit trail exceeds **the size of log files**.

Application Note: The audit trail is recorded in log file on the storage media (EMMC), the size of log file is fixed to 200K.

6.2.2 Cryptographic Support (FCS)

6.2.2.1 FCS_CKM.1/SSH/TLS Cryptographic key generation

FCS_CKM.1.1/SSH/TLS The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **algorithm supported by SSH/TLS** and specified cryptographic key sizes **key sizes supported by SSH/TLS** that meet the following: [RFC4253], [RFC4419], [RFC4432], [RFC4462], [RFC5656] for SSH/TLS.

6.2.2.2 FCS_CKM.4/SSH/TLS Cryptographic key destruction

FCS_CKM.4.1/SSH/TLS The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **overwriting with 0** that meets the following: **none**.

Application Note: Whenever a Trusted Channel is terminated for whatever reason, all temporary session keys are erased from the volatile memory by the post-processing routines associated with the Trusted Channel. These session keys are generated by FCS_CKM.1/SSH/TLS

6.2.2.3 FCS_COP.1/SSH/TLS Cryptographic operation

FCS_COP.1.1/SSH/TLS The TSF shall perform **encryption, decryption, cryptographic checksum generation for integrity and verification of checksum on TOE access channels** in accordance with a specified cryptographic algorithm **algorithm supported by SSH/TLS**, and cryptographic key sizes **key sizes supported by SSH/TLS** that meet the following: [RFC4253], [RFC4419], [RFC4432], [RFC4462], [RFC5656] for SSH/TLS.

6.2.2.4 FCS_COP.1/ROTSig Cryptographic Operation – signature verification

FCS_COP.1.1/ROTSig The TSF shall perform **verification of a digital signature** in accordance with a specified cryptographic algorithm **RSA2048** and cryptographic key sizes **2048** that meet the following: NIST SP 800-57.

Application Note: This component is used for signature verification in TOE secure booting and updating based on root of trust.

6.2.2.5 FCS_COP.1/ PubkeyHash Cryptographic Operation – hash value calculation

FCS_COP.1.1/PubkeyHash Cryptographic Operation The TSF shall perform **hash/es of public key/s used to verify the signature of a FW package** in accordance with a specified cryptographic algorithm **SHA512** and cryptographic key sizes **512** that meet the following: NIST SP 800-57.

Application Note: This component is used for hash value calculation in TOE secure booting and updating based on root of trust.

6.2.3 User Data Protection (FDP)

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

FDP_UCT.1.1 The TSF shall enforce the [access control SFP(s) for TOE users defined by FMT_SMR.1] to [transmit and receive] user data in a manner protected from unauthorized disclosure.

Application Note: This SFR ensures that the transmissions to and from the two remote users administrator and operator or the users connected to the local management network are protected from disclosure. The TOE should implement appropriate cryptographic mechanism and state of the art protocols as endorsed by the CBs. Examples are given in [BSI-1] and [SOGIS-1].

6.2.3.2 FDP_UIT.1 Data exchange integrity

FDP_UIT.1.1 The TSF shall enforce the **authenticated and authorized administrator and operators** to *transmit and receive* user data in a manner protected from **modification, deletion, insertion and replay** errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether *modification, deletion, insertion, replay* has occurred.

Application Note: This SFR ensures that the transmissions to and from the two remote users administrator and operator or the users connected to the local management network are protected from modification, deletion, replay or tampering with.

6.2.4 Identification and Authentication (FIA)

6.2.4.1 FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall detect when **5 (consecutive)** unsuccessful authentication attempts occur related to **user logging in**.

Application Note: The TSF detects the number of times the user enters the wrong password continuously, and locks the user when the maximum number of settings is reached. The number of errors can be configured, default is 5 times.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *met*, the TSF shall

1. **disable the offending user by locking the account without user notification,**
2. **Block this user from any further functionality of the TOE,**
3. **Record the event in the log file**

Application Note: When the defined number of unsuccessful authentication attempts is exceeded, the TSF terminates the user attempting to authenticate and locks the user account. The lock time can be configured, default is 5 minutes.

6.2.4.2 FIA_ATD.1 User Attribute Definition

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

1. **user ID and user name**
2. **user validity period**
3. **user level**
4. **password**
5. **password validity period**
6. **the inactivity time after which an account is automatically logged out**
7. **Status of the account (locked/unlocked)**
8. **number of failed consecutive logins within certain period of time and timestamp of last successful login**

Application Note: the attribute ‘user level’ means security roles defined in FMT_SMR.1 Security Roles.

6.2.4.3 FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [

Contains 8 to 20 characters.

Contains at least one space or the following special characters: `~!@#%&^* () - _=+\| |] ; : " ' , < > / "

Contains at least two of the following combinations: lowercase letters a-z, uppercase letters A-Z, and digits 0-9

Cannot be the same as a user name or the user name in reverse order.

]; and/or obligate the user to conduct the secret generation with appropriate means.

6.2.4.4 FIA_UAU.2 User authentication before any action

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.

Application Note: Authentication is possible by username and password.

6.2.4.5 FIA_UAU.5 Multiple Authentication Mechanisms

FIA_UAU.5.1 The TSF shall provide the following authentication mechanisms:

1. Remote authentication by LDAP

2. Local Authentication by local user name and password or SSH public key of TOE to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's identity according to the following:

1. For Remote authentication by LDAP

2. For local Authentication, the TSF will authenticate the users based on the configured Identification (including user name and password or SSH public key)

6.2.4.6 FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only **bullets(•)** or **asterisks(*)** to the user while the authentication is in progress.

6.2.4.7 FIA_UID.2 User identification before any action

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.

Application Note: Authentication is possible by username and password. The user is identified by his username if he is able to successfully authenticate with his username and corresponding password.

6.2.5 Security Management (FMT)

6.2.5.1 FMT_MOF.1 Management of Security Functions Behavior

FMT_MOF.1.1 The TSF shall restrict the ability to *modify the behavior, determine the behavior of* the functions **defined in FMT_SMF.1** to **users with sufficient user level as defined in FMT_SMR.1**.

Application Note: Access control of the TOE works as follows: All user are assigned to user roles and scope. User roles are used to control the set of commands that can be executed, and scopes are used to control the set of operating objects. Users can only execute a command if their associated user roles match the permissions of this command, and operational resources are within the scope that the user can operate on. The management of user roles also depends on this access control mechanism.

6.2.5.2 FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the **[authentication and authorization requirements defined by FMT_SMR.1 and FIA_UID.2 and FIA_UAU.2]** to restrict the ability to *[change_default, query, modify, delete]* the security attributes **[used for authorization and user role assignment]** to **[the administrator role]**.

6.2.5.3 FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **[authentication and authorization requirements defined by FMT_SMR.1 and FIA_UID.2 and FIA_UAU.2]** to provide *[restrictive]* default values for security attributes that are used to enforce the SFP.

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

6.2.5.4 FMT_MSA.4 Security attribute value inheritance

FMT_MSA.4.1 The TSF shall use the following rules to set the value of security attributes:

1. **At first startup the TOE forces the authorized administrator user to change the default security attribute values of the TOE.**
2. **The TOE user guidance shall contain recommendations for the generation of new individual security attribute values.**
3. **For the case of final TOE respectively host decommissioning, the TOE *applies methods for deletion of the stored user individual security attributes and restores the default security attribute values.***

Application Note: With respect FMT_MSA.4, rule 3: It depends on the individual case whether only the TOE, or the host, or the TOE with its host are decommissioned. Each of it could occur in case of hardware repair cases and component exchanges. In any case, also on extracted components, an abuse of stored contents should not be practical.

User individual credentials are out of control of the TOE and a compromise of those - especially after decommissioning the TOE - could implant a real threat completely out of sight and control. For that reason rule 1 of this SFR is of specific importance as it preserves the responsibility-domain separation between developer and user respectively customer.

6.2.5.5 FMT_SMF.1 Specification of Management Functions

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

1. Management of user accounts, authentication data and role association
2. Management of security banner, authentication failure policy
3. Management of customized user role and its authorities
4. Management of cypher suites
5. Management of the security audit records, audit configuration
6. Management of the session policy
7. Management of the time, NTP configuration
8. Management of TOE updating

6.2.5.6 FMT_SMR.1 Security Roles

FMT_SMR.1.1 The TSF shall maintain the roles

1. Administrator (as defined in the table below)
2. Operator (as defined in the table below)
3. Common User (as defined in the table below)
4. Custom Role(as defined in the table below)

Table 6-1 User roles

| Role | Authority |
|---------------|--|
| Common User | Users assigned with the Common User role has only the permission to modify his own password and view information, excluding OS information and operation logs. |
| Operator | Users assigned with the Operator role has all configuration and control rights, excluding user management, fault diagnosis, and security configuration. |
| Administrator | The accounts of this group are used for security management and are authorized to perform all query and configuration operations. |
| Custom Role | The administrator can define up to 4 customized roles with customized permission. |

Application Note: For roles Common user, Operator and Administrator, the roles are hierarchical, i.e. each role includes all authorities of the previous roles in addition to the authorities described for the role itself.

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 Timestamps

FPT_STM.1.1 The TSF shall be able to provide reliable timestamps.

Application Note: The reliable time stamps are based on the information of the real time clock (RTC) of the hardware. The RTC is not part of the TOE.

TOE can also connect to the NTP server and get reliable time stamps from the NTP server which is not part of the TOE.

6.2.6.2 FPT_RCV.3 Automated recovery without undue loss

FPT_RCV.3.1 After **FW package integrity failure or FW version check failure** the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2 When automated recovery from **FW package integrity- or FW version check failure** is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.3 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 configurations** 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.

Application Note: While this SFR is of general nature that no insecure state is achieved the extended SFR FPT_FLS_EXT.1 details the TOE actions and defines the secure state. The dependency to AGD_OPE.1 is automatically given by the EAL assignment.

6.2.6.3 FPT_ARB.1 Anti-Rollback

FPT_ARB.1.1 The TOE shall implement a procedure to verify the version of each FW package after its integrity has been successfully verified.

This holds also for TOE FW update packages stored by the loader mechanism.

If the version is verified the FW package is validated.

FPT_ARB.1.2 Each FW package version shall be extracted and verified with its *reference value* stored in the *OTP memory*.

An exception from this is the very initial FW package which shall be read from the ROM only.

FPT_ARB.1.3 A FW package version can have an equal or higher number as the reference number, but never a smaller number. If the version number is smaller the TOE shall stop the loading sequence and achieve a secure state.

Application Note: Executing only the correctly validated FW version is crucial for the correctness of the TSF. For that reason and if the version number is smaller than the reference number the TOE should proceed the secure state. This means for example that a smaller version number should be treated as if a FW package integrity error would have occurred. In this case the TOE shall follow then the requirements as defined in FPT_FLS_EXT.1.

6.2.6.4 FPT_FLS_EXT.1 Failure with preservation of secure state – booting

FPT_FLS_EXT.1.1 The TSF shall preserve a secure state when the following types of failures occur:

FW package integrity and FW version validation, detected after TOE FW loading, non-availability of the cryptographic coprocessor or other shortcomings.

The TOE shall run following decision-sequence to achieve either the secure state or normal operation condition:

1. Discard the FW part causing the integrity error.
2. Conduct a security event logging.
3. Conduct reloading of this FW-part from the memory source-2 storing a fallback respectively reserve image of the FW
 - a. Every restart from source-2 shall trigger according logging of the integrity fault.
 - b. After logging restart the TOE from power-off condition or continue with 5.
4. In case of successful integrity verification:
 - a. Logging of the successful verification
 - b. Achieve normal operation condition
5. Stop operation of the TOE and shutdown all services of the TOE to achieve a secure state, where only a local administrator is enabled to restart the TOE in a maintenance mode.

Application Note: It is crucial that the FW integrity is verified after loading of each part, but even before execution. For all integrity verifications during secure booting defined in FPT_ROTsb.1, in case of a detected integrity error the TOE shall achieve a secure state and never execute the faulty FW part. According to claim [CCP2], the term “secure state” refers to a state in which the TSF data are consistent and the TSF continues correct enforcement of the SFRs. A faulty executed FW part cannot ensure correct TSF enforcement.

6.2.6.5 FPT_FLS_EXT.2 Failure with preservation of secure state - update

FPT_FLS_EXT.2.1 The TSF shall preserve a secure state if the conduct of verification and validation of the new downloaded package fails or any other error occurs during the conduct of the update procedure.

The TOE shall run following sequence to achieve either the secure state or normal operation condition which is define also as secure state:

1. Trigger a TSF initiated termination of the communication channel to the update server.
2. Conduct a security event logging.
3. Discard the downloaded FW package causing the integrity error.
4. Notify the administrator user (human/non-human) from the fail and that the downloaded FW package with version information was discarded.
5. Close the FWU functionality and resume normal operation state.
6. Conduct a security event logging.
7. For the case normal operation state can no more be achieved, stop operation of the TOE and shutdown all services of the TOE to achieve a secure state, where only a local administrator is enabled to restart the TOE in a maintenance mode.
 - a. The administrator should follow a procedure in the user guidance to examine whether there is a persistent failure and

- b. should have the possibility to decide for or against a repair operation. And, if deciding against repair, the decommissioning case applies.

6.2.6.6 FPT_ROTsb.1 Root of trust secure booting procedure

FPT_ROTsb.1.1

All subsequent FW packages except of the first one, shall be verified for their integrity before execution. The integrity verification shall be conducted by digital signature verification. This requires that each single FW package comes with an according digital signature respectively certificate.

The following shall apply:

using reference data stored in the root of trust or

using trusted data which shall be hard-coded in previously integrity verified and executed FW-package.

FPT_ROTsb.1.2 The TSF shall verify the digital signature of any immutable FW code using digital signature algorithm specified in FCS_COP.1/ROTsig,

The signature verification implicitly requires the according hash calculation of the code in question. The procedures shall use the hash algorithm as specified in FCS_COP.1/PubkeyHash.

FPT_ROTsb.1.3 The TSF shall verify each certificate or public key or hash value of a public key hard coded in each FW package loaded from NVM by *calculating its hash value with hash algorithm specified in FCS_COP.1/PubkeyHash, then comparing the hash value with the public key hash within the ROT or the previous executed FW package* before the signature of the FW package is verified with these data.

FPT_ROTsb.1.4

1. During the entire procedure of secure booting the TOE does not interact with any other module or device of the Huawei server' service plane.
2. The secure booting is an offline process and shall not rely on other services.

FPT_ROTsb.1.5

1. For the case of failure and following the requirements of FPT_FLS_EXT.1, the TOE FW shall provide one fallback respectively reserve image of the TOE FW.
2. This reserve image shall be accessible for the TOE FW only during the booting phase and not during normal operation.
3. However, in case of TOE FW update it should be able to be update also the fallback image by the authorized administrator.
4. It shall be ensured that booting from this fallback or reserve image is only possible during the conduct of FPT_FLS_EXT.1.

FPT_ROTsb.1.6 The TOE shall provide a maintenance mode in case the requirements of FPT_FLS_EXT.1 are followed in fail state, and enable a locally present human administrator to enter this maintenance mode.

6.2.6.7 FPT_ROTsb.2 Root of trust secure booting atomic sequence

FPT_ROTsb.2.1

1. The TOE shall load one FW package after the other and verify its integrity with validation of the belonging digital signature or public key.
2. When the integrity of one package has been verified it is essential that the version number of this package is validated before the next package is loaded.

In case of identified failure following shall apply: If the verification of integrity of one package or the check of the version number fails, the SFR FPT_FLS_EXT.1 applies and the booting sequences is terminated therewith.

6.2.6.8 FPT_ROTsu.1 Root of trust secure updating procedure

FPT_ROTsu.1.1 The TSF shall apply data authentication and verify the integrity of a FW update package before writing it to the file system. The integrity verification shall be done by means of a digital signature. The verification reference data shall be taken from the root of trust (ROT).

FPT_ROTsu.1.2 The TSF shall verify the public key within the FW updating package by *calculating its hash value with hash algorithm specified in FCS_COP.1/PubkeyHash, then comparing the hash value with the public key hash within the ROT or the previous executed FW package* before the signature of the FW package is verified with these data.

FPT_ROTsu.1.3 The TSF shall verify the digital signature of the FW updating package using digital signature algorithm specified in FCS_COP.1/ROTSig. The TSF shall validate the version of the newly loaded TOE FW package. The interim storage of the new TOE FW package required during verification and validation procedures shall be in a TOE's NVM location not overwriting the original TOE FW image.

FPT_ROTsu.1.4 For the case of failure and following the requirements of FPT_FLS_EXT.2, the TOE FW shall stop the updating and stay with the original version.

FPT_ROTsu.1.5

1. Only when the updated TOE FW has been successfully validated and a first reboot with this updated package was successful, the former FW image shall be replaced with updated TOE FW.
2. To ensure all TSF work correctly with the new FW package the TOE shall conduct self-testing as defined with FPT_TST_EXT.1.2.
- 3.

FPT_ROTsu.1.6 The TOE shall provide a maintenance mode in case the requirements of FPT_FLS_EXT.2 are followed in fail state, and enable a locally present human administrator to enter this maintenance mode.

6.2.6.9 FPT_ROTsu.2 Root of trust atomic secure updating - conduct

FPT_ROTsu.2.1

1. All cryptographic operations conducted during the updating phase respectively the FWU operation, shall be operated on the TOE.
2. There shall be no other TOE external component in use by the FWU mechanism.

3. This means on one hand that no other host component shall be able to operate and use the TOE cryptographic coprocessor during the TOE updating sequence, and, on the other hand, that the TOE shall not make use of another non-TOE cryptographic coprocessor located on the host.

FPT_ROTUSU.2.2

1. During the entire procedure of the FWU operation and updating, the TOE does not interact with any other module or device of the GAS and service plane.
2. The FWU operation and updating shall not rely on other services such as the host HW services for random numbers, voltage, clock, and similar provisions. None of these services shall interfere on FW functional or data level.

6.2.6.10 FPT_TST_EXT.1 Basic TSF Self Testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests *during initial start-up (on power on), at the conditions during firmware updating* to demonstrate the correct operation of the TSF.

These self-testing shall comprise:

1. TOE FW integrity testing.
2. None.

6.2.7 TOE access (FTA)

6.2.7.1 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1 The TSF shall terminate an interactive session after a **time interval of user inactivity which is default or can be configured**.

Application Note: When the session is idle for more than a certain period of time, the TSF terminates the current session. For Web GUI sessions, the time interval of user inactivity is configurable by the user with administrator privileges, a minimum of 5 minutes, a maximum of 480 minutes, and a default of 5 minutes. For Redfish sessions, the time interval of user inactivity is configurable by the user with administrator privileges, a minimum of 30 seconds, a maximum of 1440 minutes, and a default of 5 minutes. For SSH and CLI sessions, the time interval of user inactivity is a default of 15 minutes and not configurable.

6.2.7.2 FTA_TAB.1 Default TOE access banners

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

Application Note: Access banner is enforced by both Web GUI and Command line interface.

6.2.7.3 FTA_TSE.1 TOE Session Establishment

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

1. **Authentication failure**
2. **User is locked**
3. **Login rules limitation based on IP address, MAC address and time segment**

6.2.8 Trusted Path/Channel (FTP)

6.2.8.1 FTP_ITC.1 Inter-TSF trusted channel

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 and disclosure.

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **identification and authentication, looking for updates of configuration parameters, looking for updates of firmware packages, sending audit records to an audit server, management from a management server.**

Application Note: The TOE use TLS to establish a secure channel to communicate with the LDAP server, and use SFTP to transfer files. Please refer to chapter 7.7 for security protocols and relative cypher suites details.

6.2.8.2 FTP_TRP.1 Trusted path

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 and replay.*

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 *initial user authentication.*

Application Note:

- a) To establish a trusted path with a remote user who utilizes WebUI or Redfish API, the TLS protocol shall be used that complies with [RFC 5246] and [RFC 8446].
- b) To establish a trusted path with a remote user who utilizes CLI, the SSH protocol shall be used that complies with [RFC 4251], [RFC 4252], [RFC 4253] and [RFC 4254].
- c) Please refer to chapter 7.7 for cypher suites for both TLS and SSH.

6.2.9 FRU_FLT.1 Degraded fault tolerance

FRU_FLT.1.1 The TSF shall ensure the operation of **the failure procedures as required by FPT_FLS_EXT.1 during the boot loading sequence** when the following failures occur: **FW package integrity and FW version validation, non-availability of the cryptographic coprocessor or other hardware shortcomings.**

6.3 TOE's FWU package

The FWU (Firmware Update) mechanism is a function of the TOE FW supported by HW means.

This update functionality shall only be accessible, after a trusted and cryptographic protected session between the TOE and the user has been established. Using such protected communication path and due to its operation in secure environments only, one focus of the evaluation of the FWU is on verification and confirmation of the fulfilment of previously defined SFRs, while the other is on the new define SFRs to protect the conduct of the update procedure.

The following set of SFRs forms a dedicated shall have completely fulfilled set of SFRs.

1. Authentication

1. FIA_UAU.2 User Authentication before any action.
2. Refined FIA_UAU.5 Multiple authentication mechanism.
3. Refined FIA_UAU.7 Protected authentication feedback.

2. Authorization

1. Refined FMT_SMF.1 Specification of management functions
2. Refined FMT_MOF.1 Management of security functions behaviour.
3. Refined FMT_MSA.1 Management of security attributes.
4. Refined FMT_MSA.3 Static attribute initialization.

3. Communication

1. Refined FTP_ITC.1 Inter-TSF trusted channel
2. FTA_SSL.3 TSF initiated termination

4. Signature verification

1. FCS_COP.1 Cryptographic operation
2. FCS_COP.1/ROTSig Cryptographic operation
3. FCS_COP.1/PubkeyHash Cryptographic operation

5. Version control

1. New defined FPT_ARB.1 Anti-Rollback

6. Conduct of FWU operation

1. New defined FPT_ROTSTU.1 Root of trust secured updating - HW support
2. New defined FPT_ROTSTU.2 Root of trust secured updating - conduct

7. Documentation and logging

1. Refined FAU_GEN.1 Audit data generation
2. FAU_GEN.2 User identity association

6.4 Security Functional Requirements Rationale

6.4.1 Objectives and Security Functional Requirements

Following table provides an overview of the assignment of SFRs to the individual objectives in order to cover those. The justification follows after the overview table.

| Objectives by CC | SFR |
|----------------------|--|
| O.Authentication | FIA_AFL.1; FIA_ATD.1; FIA_UAU.2; FIA_UAU.5; FIA_UAU.7; FIA_UID.2; FPT_STM.1; FTA_SSL.3; FTA_TAB.1; FTA_TSE.1; FTP_TRP.1; FIA_SOS.1 |
| O.Authorization | FAU_SAR.2; FAU_STG.1; FIA_ATD.1; FIA_UID.2; FMT_MOF.1; FMT_SMR.1 |
| O.FWIntegrity | FDP_UCT.1; FDP_UIT.1; FPT_RCV.3; FPT_TST_EXT.1; FRU_FLT.1; FPT_FLS_EXT.1; FPT_FLS_EXT.2; FPT_ROTSTB.1; FPT_ROTSTB.2; FPT_ROTSTU.1; FPT_ARB.1 |
| O.FWValidation | FPT_ARB.1 |
| O.Communication | FCS_CKM.1; FCS_CKM.4; FCS_COP.1; FTA_SSL.3; FTA_TSE.1; FTP_TRP.1; FTP_ITC.1 |
| O.Audit | FAU_GEN.1; FAU_GEN.2; FAU_SAR.1; FAU_SAR.2; FAU_STG.1; FAU_STG.3; FPT_STM.1 |
| O.SecurityManagement | FMT_MOF.1; FMT_SMF.1; FMT_SMR.1 |
| O.LifeCycle | FMT_MSA.4 |
| O.Ctrl_FWU | TOE'S FWU package: FIA_UAU.2; FIA_UAU.5; FIA_UAU.7; FMT_SMF.1; FMT_MOF.1; FMT_MSA.1; FMT_MSA.3; |

| Objectives by CC | SFR |
|------------------|---|
| | FTP_ITC.1; FTA_SSL.3; FCS_COP.1; FCS_COP.1/ROTSig; FCS_COP.1/PubkeyHash; FPT_ARB.1; FPT_ROTSTU.1; FPT_ROTSTU.2; FAU_GEN.1; FAU_GEN.2 |

6.4.2 Security Functional Requirements Dependency Rationale

The table below lists the SFRs used in this ST, their dependencies and whether these dependencies are covered by other SFRs in this ST. The subsequent text discusses the remaining cases.

Note that the FWU package is covered by the individual SFR dependency discussion.

| SFR | Dependency | Fulfilment by one or more SFRs in this ST |
|-----------|--|---|
| FAU_GEN.1 | FPT_STM.1 | Fulfilled |
| FAU_GEN.2 | FAU_GEN.1 FIA_UID.1 | Dependency to FAU_GEN.1 fulfilled Dependency to FIA_UID.1 see discussion below |
| FAU_SAR.1 | FAU_GEN.1 | Fulfilled |
| FAU_SAR.2 | FAU_SAR.1 | Fulfilled |
| FAU_STG.1 | FAU_GEN.1 | Fulfilled |
| FAU_STG.3 | FAU_STG.1 | Fulfilled |
| FCS_CKM.1 | [FCS_CKM.2 or FCS_COP.1] FCS_CKM.4 | Fulfilled with FCS_COP.1 and FCS_CKM.4. |
| FCS_CKM.4 | [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] | Fulfilled with FDP_ITC.1 or FCS_CKM1. |
| FCS_COP.1 | [FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1] FCS_CKM.4 | Fulfilled with FDP_ITC.1, FCS_CKM.1 or FCS_CKM.4. |
| FDP_UCT.1 | [FTP_ITC.1, or FTP_TRP.1] [FDP_ACC.1, or FDP_IFC.1] | Fulfilled with FTP_ITC.1 and FTP_TRP.1. |
| FDP_UIT.1 | None | No dependency |

| SFR | Dependency | Fulfilment by one or more SFRs in this ST |
|---------------|---|--|
| FIA_AFL.1 | FIA_UAU.1 | See discussion below |
| FIA_ATD.1 | None | No dependency. |
| FIA_SOS_EXT.1 | None | No dependency. |
| FIA_UAU.2 | FIA_UID.1 | See discussion below |
| FIA_UAU.5 | None | No dependency |
| FIA_UAU.7 | FIA_UAU.1 | See discussion below |
| FIA_UID.2 | FIA_UID.1 | Not applicable, see discussion below |
| FMT_MOF.1 | FMT_SMR.1 and FMT_SMF.1 | Fulfilled |
| FMT_MSA.1 | [FDP_ACC.1, or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1 | For FDP_ACC.1 or FDP_IFC.1 see discussion below. Fulfilled with FMT_SMR.1 and FMT_SMF.1. |
| FMT_MSA.3 | FMT_MSA.1 FMT_SMR.1 | Fulfilled with MSA.1 and FPT_SMR.1. |
| FMT_MSA.4 | FDP_ACC.1 or FDP_IFC.1 | For FDP_ACC.1 or FDP_IFC.1 see discussion below. |
| FMT_SMF.1 | None | No dependency |
| FMT_SMR.1 | FIA_UID.1 | Not applicable, see discussion below |
| FPT_ARB.1 | FAU_GEN.1 or [FDP_ACC.1, or FDP_IFC.1] | Fulfilled, see discussion in chapter 6.4.2.2 |
| FPT_FLS_EXT.1 | None | No dependency |
| FPT_FLS_EXT.2 | None | No dependency |

| SFR | Dependency | Fulfilment by one or more SFRs in this ST |
|---------------|--|--|
| FPT_RCV.3 | None | AGD_OPE.1 Operational user guidance |
| FPT_ROTsb.1 | FPT_FLS_EXT.1, FRU_FLT.1, FPT_RCV.2 | Fulfilled, see discussion in chapter 6.4.2.2 For FPT_RCV.2, ST fulfilled with FPT_RCV.3 |
| FPT_ROTsb.2 | FPT_FLS_EXT.1, FRU_FLT.1, FPT_RCV.2 | Fulfilled, see discussion in chapter 6.4.2.2 For FPT_RCV.2, ST fulfilled with FPT_RCV.3 |
| FPT_ROTsu.1 | FPT_FLS_EXT.1, FRU_FLT.1, FPT_RCV.3 FCS_COP.1, FCS_COP.1/ROtsig FCS_COP.1/PubkeyHash FPT_TST_EXT.1 Basic | Fulfilled, see discussion in chapter 6.4.2.2 For FPT_RCV.2, ST fulfilled with FPT_RCV.3 |
| FPT_ROTsu.2 | None | No dependency |
| FPT_STM.1 | None | No dependency |
| FPT_TST_EXT.1 | None | No dependency |
| FRU_FLT.1 | None | No dependency |
| FTA_SSL.3 | None | No dependency |
| FTA_SSL.3 | None | No dependency |
| FTA_TAB.1 | None | No dependency |
| FTA_TSE.1 | None | No dependency |
| FTP_ITC.1 | None | No dependency |

| SFR | Dependency | Fulfilment by one or more SFRs in this ST |
|-----------|------------|---|
| FTP_TRP.1 | None | No dependency |

6.4.2.1 Discussion of Dependencies

FAU_GEN.2 depends on FAU_GEN.1 and FIA_UID.1

The dependency to FIA_UID.1 shall not be applicable and fulfilled for this TOE as it would allow a user action before identification by the TSF. In addition, this TOE shall neither require nor enable for any TSF-mediated, non-security relevant action, before the user has been identified.

Thus, this TOE shall require without exception always identification of users, before authentication and authorization can grant access to any action of the TSF. For this reason, the dependency to FIA_UID.1 does not apply and this aspect is covered by the dedicated SFR FIA_UID2.

FDP_UCT.1 depends on FTP_ITC.1 or FTP_TRP.1, FDP_ACC.1 or FDP_IFC.1

The dependencies to FDP_ACC.1 or FDP_IFC.1 are not required for this TOE, as the access control requirement FDP_ACC.1 is substituted completely, including the enforcement, by the SFRs FIA_UID.2, FIA_UAU.2 and FMT_SMR.1. The requirement FDP_IFC.1 does not apply to this TOE, as the information flow control is covered by the atomic sequence definition given with the new defined SFR FPT_ROT.2.

The dependency to FTP_ITC.1 is fulfilled with the trusted channel established with FTP_ITC.1, as all user data are transmitted between identified, trusted endpoints in a form ensuring protection from modification and disclosure. Also, FTP_ITC.1 has been refined to cover also user data stemming from automated established server traffic, not involving human user activities.

In addition, also the dependency to FTP_TRP.1 is fulfilled, as remote (human) user data are transmitted too on a distinct communication path deploying protection from modification, replay and disclosure.

FIA_AFL.1 depends on FIA_UAU.1

The dependency to FIA_UAU.1 shall not be applicable and fulfilled for this TOE as it would allow TSF-mediated actions before the identity of the user is authenticated. In addition, this TOE shall neither require nor enable for any TSF-mediated, non-security relevant action, before the user has been identified.

Thus, this TOE shall require without exception always identification and authentication before authorization with subsequent access permission to assigned actions of the TSF can be given. For this reason, the dependency to FIA_UAU.1 does not apply and this aspect is covered by the dedicated SFR FIA_UAU2.

FIA_UAU.2 depends on FIA_UID.1

The dependency to FIA_UID.1 does not apply for this TOE as any user is required to be identified and authenticated before any action can be authorized. For this reason, the

dependency to FIA_UID.1 does not apply and this aspect is covered by the dedicated SFR FIA_UID2.

FIA_UAU.7 depends on FIA_UAU.1

The dependency to FIA_UAU.1 shall not be applicable and fulfilled for this TOE as it would allow TSF-mediated actions before the identity of the user is authenticated. In addition, this TOE shall neither require nor enable for any TSF-mediated, non-security relevant action, before the user has been authenticated. For this reason, the dependency to FIA_UAU.1 does not apply and this aspect is covered by the dedicated SFR FIA_UAU2.

FIA_UID.2 depends on FIA_UID.1

The dependency to FIA_UID.1 shall not be applicable and fulfilled for this TOE, since this TOE shall require always identification of users before authentication and authorization can grant access to any action of the TSF. For this reason, the dependency to FIA_UID.1 does not apply and this aspect is covered by the dedicated SFR FIA_UID2. This is also discussed in [CPP2] claim 990.

FMT_SMR.1 depends on FIA_UID.1

The dependency to FIA_UID.1 shall not be applicable and fulfilled for this TOE, since this TOE shall require always identification of users before authentication to a defined role and scope with authorization to defined rights can be granted. For this reason, the dependency to FIA_UID.1 does not apply and this aspect is covered by the dedicated SFR FIA_UID2. This is also discussed in [CPP2] claim 990.

FMT_MSA.1 depends on FDP_ACC.1 or FCP_IFC.1

Both dependencies are not required for this TOE, as the access control requirement FDP_ACC.1 is substituted completely, including the enforcement, by the SFRs FIA_UID.2, FIA_UAU.2 and FMT_SMR.1. The requirement FDP_IFC.1 does not apply to this TOE, as the information flow control is covered by the atomic sequence definition given with the new defined SFR FPT_ROTBS.2.

FMT_MSA.3 depends on FMT_MSA.1 and FMT_SMR.1

The dependency to FMT_MSA.1 is fulfilled as the TOE enforces the restriction to manage and treat the security attributes exclusively to the administrator role. This in turn requires the presence of a dedicated management of user roles which is provided with the provision of four hierarchical user roles each with a defined set of rights. Both dependencies are fulfilled therefore.

FMT_MSA.4 depends on FDP_ACC.1 or FDP_IFC.1

Both dependencies are not required for this TOE, as the access control requirement FDP_ACC.1 is substituted completely, including the enforcement, by the SFRs FIA_UID.2, FIA_UAU.2 and FMT_SMR.1. The requirement FDP_IFC.1 does not apply to this TOE, as

the information flow control is covered by the atomic sequence definition given with the new defined SFR FPT_ROTSTB.2.

FPT_RCV.3 depends on AGD_OPE.1

Aside from automated recovery means preserving that no FW or data is lost this SFR implements also hierarchically the option for manual recovery with human intervention. For that reason, the dependency shall be covered by the administrator's user guidance.

6.4.2.2 Discussion of Dependencies of Extended and new SFRs

FPT_ARB.1 depends on FAU_GEN.1

This TOE implements the feature to update the FW packages with appropriate protection means. Aside from the validation of each FW package updated, it may contain faults which are unknown at the point in time it is validated and installed. Execution of a validated but faulty FW could lead to undefined TOE behaviour where tracing back to the root cause of the undefined TOE behaviour could be complicated by the faulty FW package itself.

For that reason logging of the event of update, as well as rejection of an update, both are required to be logged before the update is executed respectively before the validation failure notification is given to the administrator and the invalid update FW package is discarded.

FPT_ROTSTB.1 depends on following SFRs:

- FPT_FLS_EXT.1 failure with preservation of secure state extended,
- FRU_FLT.1 degraded fault tolerance,
- FPT_RCV.2 Automated recovery

Any FW integrity check requires the availability of the cryptographic coprocessor. If on this HW path an error occurs, the requirements as defined in FPT_FLS_EXT.1 shall be followed in order to rescue the TOE from a fail state into a secure state.

Generally, if any of the HW functionalities or characteristics yields an error during this secure booting phase, there shall be only one trial to recover the TOE from fail state as defined by FPT_FLS_EXT.1. The reason is that a repetition of diversified fail states with unknown behaviour could yield vulnerability information to an attacker. This implements the requirement for FRU_FLT.1.

The dependency to FPT_RCV.3 is given intrinsically by the secure booting flow respectively by FPT_FLS_EXT.1 which both implement an automatic flow without dependency of a user action.

FPT_ROTSTB.2 depends on following SFRs:

-
- FPT_FLS_EXT.1 failure with preservation of secure state extended,
- FRU_FLT.1 degraded fault tolerance,

- FPT_RCV.2 Automated recovery

During the booting process every loaded FW package is verified for its integrity prior it is executed or installed. This verification implies also a possible fail which shall prevent on one hand that the TOE behaviour yields a vulnerability and on the other that the TOE does not provide any of its services as those are then undefined and can be faulty, leading to further problems. For that reason and in case of failure the TOE security depends on the fulfilment of the requirements of FPT_FLS_EXT.1.

The secure booting procedure involves all FW packages and depends on HW environment. This implies that several sources can have faults and errors. The TOE is therefore required to limit the tolerance of faults to zero for this booting phase. In other cases on the application layer an error trap or user information could be sufficient but the booting phase is crucial and no error can be tolerated. Therefore, the dependency to FRU_FLT.1 is given.

As discussed before the booting sequence will end up in secure state if a failure occurs. The secure state shall however first retry the booting sequence in automatic means and if failing enable an administrator user to manually recover the TOE from the secure state, which yields again the booting sequence. For that reason the dependencies to FPT_RCV.2 is justified.

FPT_ROT.1 Root of trust secured updating – HW support depends on following SFRs:

- FPT_FLS_EXT.2 failure with preservation of secure state - update,
- FRU_FLT.1 degraded fault tolerance,
- FPT_RCV.2 Automated recovery
- FCS_COP.1 Cryptographic operation
- FCS_COP.1 ROTsig Cryptographic operation
- FCS_COP.1 PubkeyHash Cryptographic operation
- FPT_TST_EXT.1 Basic TSF Self Testing

Since the update procedure itself is complex, invoking a number of other modules, and, an induction of errors during the conduct cannot be absolutely excluded, it is essential to verify that the TOE operates correctly after the new FWU package has been downloaded, validated and executed. This justifies the dependency to FPT_TST_EXT.1 with its second element being mandatory if the update functionality is part of the TOE.

Since the verification and validation procedures of the FWU have the task to identify present faults and errors before they can induce harm, it is required - as a consequence of detection - to have an according failure handling procedure in place. This justifies the dependency to FPT_FLS_EXT.2. Also, it may occur that there a HW shortcomings during the update procedures which also should not lead to harms of the TOE operations which justifies the dependency to FRU_FLT.1. However, the FWU procedure includes rebooting with the new FW package and in the worse it could be that the updated package is faulty for even unknown reason and could harm the TOE operation if not treated properly. In these cases FPT_RCV.2 preserves an automatic recovery with the original state.

Decryption, verification of the signature and hash computing requires the presence of the according cryptographic operations which depends on the SFRs FCS_COP.1, FCS_COP.1 ROTsig and FCS_COP.1 PubkeyHash.

6.4.3 Justification for SFRs covering O.Authentication

The user authentication is implemented by the user identification FIA_UID.2 followed by the authentication before any action can be conducted, FIA_UAU.2. The different authentication means for local and remote users is implemented by FIA_UAU.5. The authentication process as such is visualized only by non-information leaking signs as required by FIA_UAU.7.

The authentication process of users is based on a set of security attributes which is defined by FIA_ATD.1. If the authentication process yields a fail result, then FIA_AFL.1 provides the appropriate response to the authenticating entity.

The authentication process and results are subject of logging and entries into the log file requires a reliable time stamp which is covered by the SFR FPT_STM.1.

Authenticated, but inactive users shall face an automatic termination of the corresponding communication channel before this would implement an additional threat. The user is by that required to establish a new protected communication channel. This is covered by FTA_SSL.3. In order to inform a user about an upcoming session, the requirement FTA_TAB.1 ensures that a security banner in sense of an advisory warning message is displayed in the GUI.

Of course, the TOE TSF is enabled to deny session establishments, if the preconditions of the sessions are not met. This session rejection is addressed by FTA_TSE.1.

Remote user administrator and operator management traffic requires a segregated communication path with entity identification and initial user authentication which is covered by the trusted path FTP_TRP.1.

Since the administrator user is enabled to generate other user accounts and belonging security credentials it is essential that these security credentials conform to a certain cryptographic quality. The TOE may implement such quality testing functionality and/or the user shall be obligated by the user guidance to preserve the cryptographic quality during the security credential are created. This refined requirement fulfils FIA_SOS.1.

6.4.4 Justification for the SFRs covering O.Authorization

After successful identification and authentication the user receives a defined role and scope which are equipped with a defined set of execution rights. For that reason user identification FIA_UID.2 and the assigned user attributes as given by FIA_ATD.1 contribute and are precondition for correct authorization.

Administrators and operators conduct management operations with assigned roles and operational resources as scope which are defined by FMT_SMR.1 and managed by FMT_MOF.1.

The specific controls for the authorization for accessing and reading audit log files is covered by FAU_SAR.2, definition of permitted role and scope, and FAU_STG.1, preventing the log file from being unauthorized modified or deleted.

6.4.5 Justification for the SFRs covering O.FWIntegrity

It is obvious that FW integrity is crucial for the correct TSF. The TOE users, especially if remotely connected, send for the management activities data and FW data packages which are relevant for configuration of the TOE and TSF. If an attacker would be able to read this traffic, he could be enabled to identify FW code suitable for modification, exploit the data, or even reuse those for replay or induce malicious FW packages. This would constitute first step of an attack path. To prevent this the SFR “FDP_UCT.1 Inter-TSF user data confidentiality

protection” preserves the confidentiality of this communication between the trusted end points.

Aside from the confidentiality aspect the integrity of this data communication in context of FW package transmission is also crucial and required to meet the objective. The SFR “FDP_UIT.1 Data exchange integrity” covers this aspect.

At every booting, the very initial FW package is loaded and verified byBoorROM based on root of trust. The data root of trust is stored in immutable HW memories which is assured by OE.ImmutableROT, and the integrity of the first TOE FW is assured by OE.IntegritybyBootROM. Requirements about verification of integrity and authenticity of those FW packages are defined with the SFR “FPT_ROTSTB.1 Root of trust secure booting procedure” and are required to meet the objective O.FWIntegrity.

Furthermore, the SFR “FPT_ROTSTB.2 Root of trust secure booting atomic sequence” preserves the integrity verification of each single loaded FW package in autarkic and non-interruptible ways and without undiscovered modifications from outer TOE or host environment. This covers the objective O.FWIntegrity.

It can for whatever reason occur that an integrity verification of FW package fails during its loading when booting. In this cases it is essential that on one hand the faulty FW package is not executed and on the other hand that the TOE achieves a secure state. This cannot be handled by trap and user action, as the TOE is still in the booting phase. For that reason the SFRs “FPT_FLS_EXT.1 Failure with preservation of secure state – extended” and “FRU_FLT.1 Degraded fault tolerance” are essential that only integrity verified FW packages are executed and that the TOE follows the therein defined requirements in case of failure. This covers the objective O.FWIntegrity.

Nevertheless, even if an FW package integrity failure occurs, the TOE shall have the option to recover from the achieved secure state in defined ways and with integrity verification of the loaded FW packages from another TOE internal source. These recovery paths are hierarchically defined in the SFR “FPT_RCV.3 Automated recovery without undue loss”. As such recovery resolves integrity failures from the first source and provide integrity protected loading from a second source the objective O.FWIntegrity is met.

The SFR “FPT_TST_EXT.1 TSF testing” shall ensure on one hand that the ROT basis for correct integrity verification of the FW packages is working correctly and on the other hand that the FW packages itself are configurable tested during runtime. Both types of testing protect the FW integrity from persistent faults in HW and ephemeral, accidental induced faults at runtime, for example voltage glitches or anything similar. This covers the objective O.FWIntegrity.

6.4.6 Justification for the SFRs covering O.FWValidation

Validation of each single FW package FW means that its correctness is confirmed and that it is ready for execution. Correctness means on one hand that its integrity has been verified and after this that the correct FW package version has been loaded. This prevents that not an old, possibly vulnerable - but integrity verified – version comes to execution.

The SFR “FPT_ARB.1 Anti-Rollback” defines first the sequence of testing – first integrity and then version test - which leads then in consequence to the validation of the FW package. This means also that the corresponding SFRs covering O.FWIntegrity are contributing to the O.FWValidation as well.

6.4.7 Justification for the SFRs covering O.Communication

Any communication between the TOE and an external entity including the entities on the local management network is protected by secured protocols. The selection of these protocols shall ensure that a) an inter-TSF channel, as distinct channel from other traffic is established to conduct the TOE functions. And, b) a logical distinct trusted path to remote users is generated. This covers FTP_ITC.1 and FTP_TRP.1.

The developer shall select symmetric, asymmetric, authentication and integrity cryptographic mechanisms from endorsed standards which covers then sufficiently FCS_COP.1, FCS_CKM.1 and FCS_CKM.4. This ensures that the cryptography used is publicly discussed and free of mathematical characteristics which could be exploited easily. However, there is never a guarantee for the absence of inherent vulnerabilities, but as long following the recommendation from the CBs, there was none identified during the years of publication and research on it.

Authenticated, but inactive users shall face an automatic termination of the corresponding communication channel before this would implement an additional threat. For example, the authenticated has left the working place without locking it and another person abuses the given communication path. The SFR ensures a configurable termination and in consequence, the user is by that required to establish a new protected communication channel. This is covered by FTA_SSL.3.

Of course, the TOE TSF is enabled to deny session establishments, if the preconditions of the sessions are not met. This session rejection is addressed by FTA_TSE.1.

6.4.8 Justification for the SFRs covering O.Audit

The generation of audit records of defined and configurable events is essential for administration purposes, for example tracing back to a root cause of failure or responsible user.

The information and data at least to be logged is implemented by FAU_GEN.1.

The audit process and log file entries require a reliable time stamp which is covered by the SFR FPT_STM.1 which needs to be combined with the information of user identity association which is covered by FAU_GEN.2.

Log file can contain critical data such as user identities and can fall under the GDPR regulation. For that reason reading shall only be allowed to a dedicated user role and therein to administration users having the explicit assigned right to do so. These requirements are defined in FAU_SAR.1 and FAU_SAR.2.

The integrity protection of the stored audit records against unauthorized modification, which could impair the purpose of the log file but could also violate the GDPR if personal data are affected, is implemented in FAU_STG.1.

For the case the audit trail respectively log file exceeds the size of the storage device, the TSF shall begin overwriting the oldest records. This ensures that a time synchronized sequence of most recent events is given any time the log file is read. This fulfils FAU_STG.3.

6.4.9 Justification for the SFRs covering O.SecurityManagement

The TOE organization of security attribute based identification, authentication and authorization requires a management of the related functionality and a configuration thereof. Therefore, the security management shall provide a defined set of security management functions, which is defined in the requirement FMT_SMF.1. Specifically, the configuration of

this set of functions shall only be conducted by the permitted user which is covered by FMT_MOF.1. The user permission by the role is defined in FMT_SMR.1.

6.4.10 Justification for the SFRs covering O.LifeCycle

The rule definition in FMT_MSA.4 covers several aspects treating the separation of security attribute responsibilities and their treatment affecting the initial setting into operation and decommissioning of the TOE.

When setting the TOE first time into operation, the individualization of the security attributes preserves on one hand a separation of responsibilities for the security attribute domains, and on the other hand a protection of the TOE for the case the default security attributes get compromised.

At the lifecycle phase end of life or decommissioning, the fulfilment of the requirement preserves that a TOE abuse by malicious identification and authentication due to compromised user individual credentials is rendered not practical.

6.4.11 Justification for the SFRs added by O.Ctrl_FWU

Since the protection of the TOE FW update is of paramount importance for the correctness of the GAS the corresponding tool to conduct, the FWU mechanism, has this importance as well.

The evaluation shall therefore explicitly verify and confirm all the listed SFRs building the package for the FWU protection covering authentication, authorization, communication, cryptographic support and version control are fulfilled. It is crucial that all SFRs for the FWU package are fulfilled.

Discussion in detail

Conducting a TOE FW update or patching is required to be conducted only and exclusively by the defined user role or even person. This requires strong authentication means with the right tools which are covered by FIA_UAU.2 User Authentication before any action, refined FIA_UAU.5 Multiple authentication mechanism and refined FIA_UAU.7 Protected authentication feedback. When authenticated it is essential that the correct role is equipped the assigned rights to access and to execute. This shall be based on security attributes and clear management which is covered by the FMT_SMF.1 Specification of management function, including the optional function to conduct TOE's FWU, the FMT_MOF.1 Management of security functions behaviour, FMT_MSA.1 Management of security attributes and FMT_MSA.3 Static attribute initialization which have all been refined. These SFR define the functions, their behaviour, and their attributes with initialization enabling for authorizing the user to conduct the desire TSF including the option FWU.

Due to proven best practice and risk mitigation for the management traffic the by the SFR defined TOE user traffic shall be operated on a distinct communication path separated from traffic handled on the GAS's service plane. It is also essential that this communication path preserves the integrity and confidentiality of the transmitted user data. These partial objectives are covered by the refined FTP_ITC.1 Inter-TSF trusted channel.

Then aside from the protection from the endpoints and during the transmission, the payload itself shall be verified for its integrity as the payload could have faults or was tampered even before transmission. For that reason each package shall be verified by digital signature on it. This verification, including steps for decryption and hash computation with comparison, is covered FCS_COP.1 Cryptographic operation.

Then it could be for unknown reason on remote user's side that the distinct communication path is kept open although it is not used anymore. To prevent from any abuse possibility the

TOE can initiate the termination of this unserved path which is covered by FTA_SSL.3 TSF initiated. Termination.

Another source of accidentally fault or attacking threat is the TOE FW update with an obsolete or malicious version, for example with publicly known vulnerabilities, which could pass all previous protection means and even its signature verification. All that and some adverse use cases more described in the security problem chapter are countered, if the defined set of SFRs constituting the FWU SFR package is completely fulfilled. This comprehensive set of SFRs covers the security functional requirements of authentication, authorization, communication protection, cryptographic operations with iterations, version validation, protective conduct of operation and its documentation. To ensure the FWU protection the SFRs FPT_ARB.1, FPT_ROTUSU.1 and FPT_ROTUSU.2 were newly defined. The other SFRs have been previously defined as given in the FWU SFR package and complete the update mechanism protection.

The objective is met by the FWU SFR package.

6.5 Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level 4 with augmentation ALC_FLR.1, as specified in [CC] Part 3.

6.6 Security Assurance Requirements Rationale

The evaluation assurance level 4 with augmentation ALC_FLR.1 has been chosen commensurate with the threat environment that is experienced by typical consumers of the TOE.

Dependencies within the EAL package selected (EAL4) for the security assurance requirements have been considered by the authors of CC Part 3 and are therefore not analyzed here.

7 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

7.1 Security Audit

- The TOE provides an audit trail consisting of operation logs and security logs: support recording non-query operations in the operation logs, security-relevant operations in the security logs, the TOE generation of audit logs for the following events:
 - i. User activity
 1. login, logout
 2. add, delete, modify users
 3. user password change
 4. user level change
 5. user lock and unlock
 - ii. System management
 1. perform security management functions which are defined in FMT_SMF.1 Specification of Management
 2. Power on, power off, and restart the service system
- Operation logs and Security logs record the following information: the operation interface (if applicable), access IP address (if applicable), date and time, the outcome, and subject identity (if applicable), for all audit events the corresponding timestamp will be recorded together with the event.
- Users with security configuration rights can query and download operation logs and security logs, can review that whoever accesses and logins the system and any operation on the system according to the content of the security log and the operation log.
- iBMC logs are saved in the EMMC flash file system in real time, the logs will not be lost when the system is powered off or restarted, nor be modified by any user. None of TOE user has the privilege to write/modify/delete any TOE log.
- The operation logs and security logs keep records in time sequence. When the size of the log file reaches the specified size, the log file is automatically backed up.
- iBMC can also transfer logs to a syslog server in real time. You can configure a syslog server to restore logs to it.

(FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_STG.1, FAU_STG.3, FPT_STM.1)

7.2 Cryptographic functions

Cryptographic functions are required by security features as dependencies. These encryption algorithms protect the confidentiality and integrity of key data at transmission or storage. For cryptographic algorithms used for trusted path/channel, please refer to chapter 6.7.

For secure booting and updating based ROT, the TOE support signature verification using following algorithms:

- Hash: SHA512 meets NIST SP 800-57,
- Digital signature: RSA2048 meets NIST SP 800-57.

(FCS_CKM.1/SSH/TLS, FCS_CKM.4/SSH/TLS, FCS_COP.1/SSH/TLS, FCS_COP.1/ROTSig, FCS_COP.1/PubkeyHash)

7.3 Identification and Authentication

The TOE can identify users based on unique IDs and enforce their authentication before granting them access to any TSF management interfaces. Detailed functions include:

- Support for user individual attributes including the user ID, user level, and password to ensure that each user is unique in the system. TOE record user name, password, and user level for each local user. User privilege determine which TOE functions can be used, user privilege include user management, basic management, remote control, VMM, security management, power control, diagnostics, query and own password & SSH configuration.
- The TOE support authentication via local username and passwords. This function is achieved by comparing user information input with pre-defined user information stored in the flash. Users must successfully identify and authenticate before they are allowed to use TOE functions. The TOE enforces a password complexity :

Contains 8 to 20 characters.

Contains at least one space or the following special characters: `~!@#\$\$%^& () -_ =+||[];: ", <, > /"*

Contains at least two of the following combinations: lowercase letters a-z, uppercase letters A-Z, and digits 0-9

Cannot be the same as a user name or the user name in reverse order.

- Support authentication via the remote LDAP authentication server. The TOE hands identification and authentication information provided by the user during login to the LDAP server and enforces the LDAP server's pass/fail decision.
- The password entered by the user is not displayed in plaintext during authentication. The password is displayed as asterisks (*).
- Support logout when no operation is performed on the user session within a specified interval. If an account that has logged in does not exchange information with the TOE within the specified interval, it will be automatically logged out. The account needs to be authenticated again for a new login.

- Support maximum attempts for authentication failures. By default, after five consecutive login attempts using one account fail, the account is locked. A log is recorded after the account is locked. The default value of lock period is 5 minutes, the user account will be automatically unlocked after 5 minutes by default.
- The TOE controls access by the group-based authorization framework with predefined role groups (also called user level) for management. Four hierarchical access groups are offered and can be assigned to individual user accounts.
- Only authenticated users can execute commands of the TOE. Only one user level can be assigned to a user account. So the user level of a user is unambiguous at any time. All authenticated users of the TOE are administrative users of some kind belonging to one of the user levels defined below. There are no authenticated non-administrative users.
- Accounts are managed in groups and each group represents a specific authority assigned to the accounts in the group. The table below lists the groups and their definition. For example, the accounts of the "administrator" group are authorized to perform all security management and advanced diagnosis operations. When an account is created, it is authorized to perform certain operations and is not allowed to perform unauthorized operations.

Table 7-1 Groups of accounts

| Group | Authority |
|---------------|---|
| Administrator | The accounts of this group are used for security management and are authorized to perform all query and configuration operations. |
| Operator | Users assigned with the Operator role has all configuration and control rights, excluding user management, fault diagnosis, and security configuration. |
| Common User | Users assigned with the Common User role has only the permission to modify his own password and view information, excluding OS information and operation logs. |
| Custom Role | The system administrator defines the permission of custom users as required. The iBMC supports a maximum of four custom users. System permissions consist of common settings, remote control, remote media, security configuration, power control, diagnosis, query permissions, and configuration. Administrators can set any combination of them as a custom role's permission. |

(FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.5, FIA_UAU.7, FIA_UID.2)

7.4 Security Management

The TOE offers management functionality for its security functions.

- The security functions include:
 - ✓ Management of user accounts, authentication data and role(level) association
 - ✓ Management of security banner, authentication failure policy
 - ✓ Management of customized user role and its authorities
 - ✓ Management of cypher suites

- ✓ Management of the security audit records, audit configuration
- ✓ Management of the session policy
- ✓ Management of the time, NTP configuration.

There are four hierarchical user levels: Administrator, operator, common user and custom user as described in chapter 6.3 Identification and Authentication

- A user level is assigned to each account, so that user accounts are managed by levels. When an account is created, it is authorized to perform certain operations and is not allowed to perform unauthorized operations according to which user level it is associated with.
- Before successfully identified and authenticated, a user can't perform any action to the TOE.
- In order to prevent possible privilege escalations, users can change user attributes (esp. their own attributes) only for users up to their own user level and cannot increase the user level attribute beyond their own user level.
- Only the administrator has the permission to configure all user, including password change and permission modification. Users of other roles can only configure themselves.
- To ensure security isolation between vendor and TOE user, or between TOE administrator and a new user account it creates, a TOE user must modify its password (shall meets complexity requirements) at first login.

(FMT_MOF.1, FMT_MSA.1, FMT_MSA.3, FMT_MSA.4, FMT_SMF.1, FMT_SMR.1)

7.5 Protection of the TSF

The integrity and authenticity of the TOE FW is protected based on hardware root of trust (ROT), which contains immutable trust data (hash of root public key). Both FW booting and updating must not be executed unless passing verification based on these data.

During TOE booting, the first FW package of the TOE which loading by BootROM, 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. The public key authentication is according to public key hash which stored in the immutable eFuse of iBMC hardware. 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. Therefor the integrity and authenticity of the FW is assured.

During updating, the TOE will verify the authenticity and integrity of the new version package by verify its digital signature with authenticated public key according to public key hash which stored in the immutable eFuse of iBMC hardware. 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 mode 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 recovered to a default setting to maintain security.

The TOE implements Anti-rollback mechanism by verifying the validity of a version according to relevant flag bits in the effuse (an OTP component) provided by iBMC hardware,

(its operational environment) . While updating to a new FW image which invalidate another older version, the TOE will set the old FW version as invalid by burning relevant info to the efuse, therefore prevent rolling back to this version in the future.

(FPT_RCV.3, FPT_ARB.1, FPT_FLS_EXT.1, FPT_FLS_EXT.2, FPT_ROTSTB.1, FPT_ROTSTB.2, FPT_ROTSTU.1, FPT_ROTSTU.2, FPT_TST_EXT.1, FRU_FLT.1)

7.6 TOE Access

The session timeout period can be configured on the iBMC. If you do not perform any operation for a long time, the session will be automatically disconnected. You can also configure warning information on the iBMC login page to prompt all login users. If a user fails to log in to the system for multiple times, the user is locked. The number of login failures and the locking duration are configurable.

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: A user initiates a request to terminate a session. The system administrator can terminate sessions initiated by other users.

The TOE supports configuration of default access banner. Before user authentication, warning information is displayed on the page. The warning information can be configured by users. The TOE supports the anti-brute force cracking mechanism, Accounts can be locked based on consecutive login failures.

The iBMC software supports scenario-based login restriction, and access control policy based on time range, IP address, and MAC address. Only the users complying with the configured login time, IP address segment, or MAC address segment are allowed to access the system through the management channel. In this way, the access control of the server management interface is controlled within the minimum range.

(FTA_SSL.3, FTA_TAB.1, FTA_TSE.1)

7.7 Trusted Path/Channel, User Data Protection

The TOE provides communication security by implementing trusted channels/paths using TLS or SSH communication protocol.

For TLS, the TOE acts as a TLS server and allows a trusted entity to initiate communication. The TOE has been loaded with a preset TLS certificate as a default one before delivery. Authorized user can manage TLS certificates such as querying information and renewing certificates. It is strongly recommended for user to replace the default certificates with a new one.

The TOE implements TLS 1.3 ([RFC 8446]) and TLS 1.2 ([RFC 5246],) for following communications:

- User access via WebUI
- User access via Redfish API
- Interaction with LDAP server

- Interaction with syslog server

And SSH V2.0 ([RFC 4251], [RFC 4252], [RFC 4253], [RFC 4254]) for:

- Remote user access via CLI
- File transfer to temp directory through SFTP

To ensure high security strength, the TOE using following cipher suites for them respectively.

Cipher suites for TLS 1.3 and TLS 1.2:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC5288
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC5288
- TLS_DHE_DSS_WITH_AES_256_GCM_SHA384 as defined in RFC5288
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC5288
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC5288
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC5288
- TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 as defined in RFC5288
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC5288
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 as defined in RFC7905
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 as defined in RFC7905
- TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256 as defined in RFC7905
- TLS_DHE_RSA_WITH_AES_128_CCM as defined in RFC6655
- TLS_DHE_RSA_WITH_AES_256_CCM as defined in RFC6655
- TLS_ECDHE_ECDSA_WITH_AES_128_CCM as defined in RFC7251
- TLS_ECDHE_ECDSA_WITH_AES_256_CCM as defined in RFC7251
- TLS_CHACHA20_POLY1305_SHA256 as defined in RFC8446
- TLS_AES_256_GCM_SHA384 as defined in RFC8446
- TLS_AES_128_GCM_SHA256 as defined in RFC8446

Cipher suites for SSH V2.0:

| usage | Algorithm | Standard |
|-------------------------------|------------|----------|
| Encryption for session | aes128-ctr | RFC-4344 |
| | aes192-ctr | RFC-4344 |
| | aes256-ctr | RFC-4344 |

| | | |
|--|--------------------------------------|----------|
| | aes128-gcm@openssh.com | RFC-5647 |
| | aes256-gcm@openssh.com | RFC-5647 |
| | chacha20-poly1305@openssh.com | RFC-7905 |
| key agreement for session establishment | curve25519-sha256 | RFC-8731 |
| | curve25519-sha256@libssh.org | RFC-8731 |
| | diffie-hellman-group-exchange-sha256 | RFC-4419 |
| MACs | hmac-sha2-512 | RFC-6668 |
| | hmac-sha2-512-etm@openssh.com | RFC-6668 |
| | hmac-sha2-256 | RFC-6668 |
| | hmac-sha2-256-etm@openssh.com | RFC-6668 |
| Server authentication | ssh-ed25519 | RFC-8709 |
| | rsa-sha2-256 | RFC-8332 |
| | rsa-sha2-512 | RFC-8332 |

For all communications between the TOE and any remote user, the TOE provides integrity and confidentiality protection for user data by trusted path including SSH or HTTPS. The TOE supports session time-out after a configurable time of user inactivity. After the session has expired, the equipment user account will be automatically logged out.

The TOE supports denying session establishment based on authentication failure (i.e. device authentication failure for TLS and user authentication as well as device authentication failure for SSH).

(FTP_ITC.1, FTP_TRP.1, FDP_UCT.1, FDP_UIT.1)

A Abbreviations, Terminology and References

A.1 Abbreviations

| | |
|-----------|--|
| BMC | Baseboard Management Controller |
| CC | Common Criteria |
| CM or CMS | Configuration Management or Configuration Management System |
| DDoS | Distributed Denial of Service |
| DoS | Denial of Service |
| EAL | Evaluation Assurance Level |
| GUI | Graphical User Interface (usually a browser based visualization) |
| HSM | HW Security Module |
| IC | Integrated Circuit |
| IEC | International Electrotechnical Commission |
| IPMI | Intelligent Platform Management Interface |
| ISCI | International Security Certification Initiative hosted and operated from Eurosmart |
| LDAP | Lightweight Directory Access Protocol |
| NTP | Network Time Protocol |
| NVM | Non Volatile Memory |
| OTP | One-Time-Programmable memory |
| PP | Protection Profile |
| RNG | Random Number Generation |
| RMCP | Remote Management Control Protocol |
| RMT | Remote Management Terminal |
| SAR | Security Assurance Requirement |
| SFR | Security Functional Requirement |
| SFTP | Secure File Transfer Protocol |

| | |
|-------|---|
| SNMP | Simple Network Management Protocol |
| SOGIS | Senior Officials Group Information Systems Security |
| SSH | Secure Shell |
| ST | Security Target |
| TLS | Transport Layer Security |
| TOE | Target of Evaluation |
| TSF | TOE Security Functionality |

A.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

Administrator: An administrator in the content is the user of the TOE who may have been assigned specific administrative privileges within the TOE.

User: A user is a human or a product/application using the TOE.

A.3 References

- [BSI-1] BSI, Cryptographic Mechanisms: Recommendations and Key Lengths, BSI TR-02102-1, 2020-01 as of 2020-03-24
- [CC] Common Criteria for Information Technology Security Evaluation, Part 1-3, Version 3.1 Revision 5, April 2017
- [CCP1] Common Criteria for Information Technology, Security Evaluation, Part 1: Introduction and general model, April 2017, Version 3.1 Revision 5
- [CCP2] Common Criteria for Information Technology, Security Evaluation, Part 2: Security functional components, April 2017, Version 3.1 Revision 5
- [CCP3] Common Criteria for Information Technology, Security Evaluation, Part 3: Security assurance components, April 2017, Version 3.1 Revision 5
- [CEM] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1 Revision 5, April 2017
- [FIPS 180-4] FIPS PUB 180-4 – Secure Hash Standard (SHS), August 2015
- [FIPS 186-4] FIPS PUB 186-4 – Digital Signature Standard (DSS), July 2013
- [FIPS 197] FIPS PUB 197 – Advanced Encryption Standard (AES), November 26, 2001
- [FIPS 198-1] FIPS PUB 198-1 - The Keyed-Hash Message Authentication Code (HMAC), July 2008
- [IEC-1] Security for industrial automation and control systems – Part 4-2: Technical security requirements for IACS components, edition 1.0 as of 2019-02, IEC 62443-4-2

- [NIST SP 800-38A] NIST Special Publication 800-38A – Recommendation for Block Cipher Modes of Operation: Methods and Techniques, December 2001
- [NIST SP 800-38D] NIST Special Publication 800-38A – Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007
- [NIST SP 800-56A] NIST Special Publication 800-56A Rev.3 – Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, April 2018
- [NIST SP 800-56B] NIST Special Publication 800-56B Rev. 2 – Recommendation for Pair-Wise Key-Establishment Schemes Using Integer Factorization Cryptography, March 2019
- [PKCS#1 V2.1] PKCS #1 v2.1: RSA Cryptography Standard, June 2002
- [PKCS#3] PKCS #3: Diffie-Hellman Key- Agreement Standard, version 1.4, November 1993
- [RFC 2104] RFC 2104 - HMAC: Keyed-Hashing for Message Authentication, February 1997
- [RFC 3174] RFC 3174 - US Secure Hash Algorithm 1 (SHA1), September 2001
- [RFC 3268] RFC 3268 - Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS), June 2002
- [RFC 3526] RFC 3526 - More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE), May 2003
- [RFC 4250] RFC 4250 - The Secure Shell (SSH) Protocol Assigned Numbers, January 2006
- [RFC 4251] RFC 4251 - The Secure Shell (SSH) Protocol Architecture, January 2006
- [RFC 4252] RFC 4252 - The Secure Shell (SSH) Authentication Protocol, January 2006
- [RFC 4253] RFC 4253 - The Secure Shell (SSH) Transport Layer Protocol, January 2006
- [RFC 4254] RFC 4254 - The Secure Shell (SSH) Connection Protocol, January 2006
- [RFC 4344] RFC 4344SSH Transport Layer Encryption Modes January 2006
- [RFC 4346] RFC 4346 - The Transport Layer Security (TLS) Protocol Version 1.1, April 2006
- [RFC 4419] RFC 4419 - SSH DH Group Exchange, March 2006
- [RFC 4432] RFC 4432 - SSH RSA Key Exchange, March 2006
- [RFC 4634] RFC 4634 - US Secure Hash Algorithms (SHA and HMAC-SHA), July 2006
- [RFC 4462] RFC 4462 - SSH GSS-API Methods, May 2006
- [RFC 5246] RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2, August 2008

- [RFC 5288] RFC 5288 - AES Galois Counter Mode (GCM) Cipher Suites for TLS, August 2008
- [RFC 5487] RFC 5487 - Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode, March 2009[RFC 5647] RFC 5647 - AES-GCM for Secure Shell, August 2009
- [RFC 5656] RFC 5656 - SSH ECC Algorithm Integration, December 2009
- [RFC 6234] RFC 6234 - US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF), May 2011
- [RFC 6655] RFC 6655 - AES-CCM Cipher Suites for Transport Layer Security (TLS), July 2012
- [RFC 6668] RFC 6668 - Sha2-Transport Layer Protocol, July 2012
- [RFC 7251] RFC7251 - AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS, June 2014
- [RFC 7905] RFC 7905 - ChaCha20-Poly1305 Cipher Suites for Transport Layer Security (TLS), June 2016
- [RFC 8017] RFC 8017 - PKCS #1: RSA Cryptography Specifications Version 2.2, November 2016
- [RFC 8332] RFC 8332 - Use of RSA Keys with SHA-256 and SHA-512, March 2018
- [RFC 8446] RFC 8446 - The Transport Layer Security (TLS) Protocol Version 1.3, August 2018
- [RFC 8492] RFC 8492 - Secure Password Ciphersuites for Transport Layer Security (TLS), February 2019
- [RFC 8709] RFC 8709 - Ed25519 and Ed448 Public Key Algorithms for the Secure Shell (SSH) Protocol, February 2020
- [RFC 8731] RFC 8731 - Secure Shell (SSH) Key Exchange Method Using Curve25519 and Curve448, February 2020
- [SOGIS-1] SOG-IS Crypto Working Group, SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms, version 1.2, 2020-January