# SECURITY TARGET

## (CC EAL2)

## COSMICGUARD FOR LINUX (GMV)


| | |
|---|---|
| Prepared by: | Daniel Castro de la Rosa<br>José Jorge García Almajano |
| Verified by: | Daniel Castro de la Rosa |
| Approved by: | Pedro Celis de la Hoz |
| Authorized by: | Juan Jesús León Cobos |



| | |
|---|---|
| Code: | GMV-COSMICGUARD-ST |
| Date: | 2025/11/18 |
| Version: | 7 |
| Internal code: | GMV 22617/24 V1/24 |

# DOCUMENT CHANGE CONTROL

| Version | Date | Pages | Processor | Changes |
|---|---|---|---|---|
| 1 | 2024/04/30 | 49 | Word 365 | Initial document |
| 2 | 2025/01/22 | 51 | Word 365 | Reviewed with CPSTIC comments:<br>• TOE name modified<br>• Added FIA_SOS.1<br>• Added FTA_SSL.3<br>• Application note added on FAU_SAR.2<br>• FMT_SMF.1 modified<br>• Other minor changes on the ST Introduction and The TOE Summary Specification |
| 3 | 2025/02/27 | 51 | Word 365 | Reviewed with CPSTIC comments:<br>• Typo corrected on FDP_SDI.2.2 title<br>• Added operations on FDP_ACC.1(1)<br>• Added subjects, objects and operations on FDP_ACC.1(2)<br>• User roles aligned between FMT_SMR.1 and TSS |
| 4 | 2025/04/14 | 54 | Word 365 | Reviewed with ORs:<br>• Non-TOE components reorganized and clarified<br>• Added table of definitions and acronyms<br>• Physical scope of the TOE rewritten<br>• Added TOE delivery procedure<br>• Added SFRs rationale for the security objectives<br>• Added convention for operations on SFRs<br>• Asset description added, threats rewritten to better describe assets<br>• Security objectives and OSPs modified |
| 5 | 2025/06/02 | 55 | Word 365 | Reviewed with ORs:<br>• Physical scope of the TOE modified<br>• Evaluated configuration updated with newer versions<br>• FAU_SAR.1 modified<br>• Application note added on FIA_SOS.1 |
| 6 | 2025/07/18 | 55 | Word 365 | Reviewed with ORs:<br>• Application note added on FIA_SOS.1<br>• Typos corrected in the ST introduction |
| 7 | 2025/11/18 | 55 | Word 365 | Reviewed with ORs:<br>• Information classification level defined as public<br>• Document references clarified<br>• Updated TOE checksum |

# TABLE OF CONTENTS

# LIST OF TABLES AND FIGURES

# 1. ST INTRODUCTION

## 1.1. ST REFERENCE

| | |
|---|---|
| **Title** | CosmicGuard for Linux. Security Target |
| **Sponsor** | GMV Innovating Solutions S.L. |
| **Author(s)** | Daniel Castro de la Rosa<br>José Jorge García Almajano |
| **ST Version** | 7 |
| **ST Publication Date** | 2025/11/18 |
| **CC Version** | 3.1 Release 5 |
| **Assurance Level** | EAL2 |

Table 1: Security Target reference

## 1.2. TOE REFERENCE

The target of evaluation (TOE) in this Security Target (ST) is CosmicGuard for Linux (version 2.2) developed by GMV Innovating Solutions S.L.

# 1.3. TOE OVERVIEW

## 1.3.1. TOE TYPE AND SCOPE

The TOE described in this ST is a software product designed to protect Linux endpoints (Agent) and managing the Access-Control-List (ACL) and the audit from a centralized web console (Server). The TOE employs this ACL to control the execution of applications within the endpoint and prevent any unauthorized access. This is achieved by enforcing the rules contained in this ACL.

The TOE provides security functionality, it includes functions such as Endpoint Protection, Management and Audit.

It falls under the category of Other Devices and Systems, as identified on the Common Criteria Portal (www.commoncriteriaportal.org), which lists all certified products.

## 1.3.2. TOE USAGE AND MAJOR SECURITY FEATURES

### 1.3.2.1. TOE USAGE

Cyber-attacks have increased significantly since 2020, with research institutions and government organizations being particularly vulnerable due to the sensitive information they hold. Protecting against these threats is a growing concern across all industries. In critical environments, traditional solutions such as antivirus and antimalware may not provide sufficient security as they are designed for a different type of user.

GMV has broad experience in the field of whitelisting technology, having achieved notable success with Checker for ATMs, which currently helps numerous banking institutions worldwide to protect their ATMs.

The TOE is an endpoint solution specifically designed for Satellite Control Centers that allows organizations to protect their critical assets.

To provide some context regarding the environment in which the TOE is deployed, it is necessary to highlight that due to the criticality of the service provided by a Satellite Control Centre, it is important to note that these are environments with high security measures. This is evidenced by the fact that connections to the outside are very limited and controlled, and access from the outside must be done via VPN connections. Physical access to the areas where the most critical services are provided must be protected by the implementation of rigorous security measures, with the objective of preventing unauthorized personnel from gaining access.

Further protective measures include the segmentation and protection of internal networks by intermediate firewalls, which serve to isolate networks and monitor data transmission. Furthermore, operators perform their work on endpoints that have been bastioned in accordance with the client's policies, and they are required to authenticate in order to make use of the system.

At client's premises it is needed to have PKI systems in order to generate certificates, thereby ensuring the security of connections and, if required, the mutual authentication of the parties involved.

The endpoint is the asset where the TOE is deployed and what is protected. This is performed by applying an ACL, which allows only and exclusively the rules included in it.

### 1.3.2.2. TOE SECURITY FEATURES

The TOE comprises two components: a centralized server managed through a web console and the client that protects the endpoint against the execution of unwanted applications by applying the rules in an ACL.

The ACL can be built by a learning mechanism that automatically populates the rules or by including rules manually. It is recommended that this process be conducted in a test environment. Once the process is complete, the ACL can be sent to the target endpoint in the production environment, since then the rules will be applied immediately.

In terms of security, the features of the TOE are:

- **Fine-grained control over process execution**.
  The control implemented is based on blocking the execution of the processes which are not listed in the rules of the ACL and allowing its execution based on the path of the executable or its signature, in addition to the permissions with which it is launched and the parameters it executes.

- **Control over loaded libraries.**
  Access control to load libraries is restricted to those processes considered essential by the ACL rule, and only in read mode.

- **File resources protection.**
  File resource protection is based on restricting access to those files considered most critical only by those processes listed in the ACL rules in a number of modes, e.g. read/write, or read-only mode, etc…

- **Control over inbound and outbound network connections.**
  It is possible to restrict incoming and outgoing connections by basing the ACL rule on an IP address or a range of them, as well as a port or a range of them. Furthermore, it is possible to filter by protocol type, including TCP, UDP, or others.

- **Centralized management console.**
  Access to the console is web-based and via HTTPS connection, which allows centralized management of all the functionality of the solution, thus isolating the administration of the system.

- **Identification and authentication of console users and role management.**
  The user that accesses to the management web console is identified and authenticated. A wide range of roles can be assigned for the purpose of providing fine-grained control over the actions that can be performed on the console.

- **Audit review of ACL violations.**
  It is possible to determine the nature of the ACL violations at the endpoints.

- **Audit of user actions.**
  All actions performed by users within the administration web console are subject to audit for subsequent review.

The following table highlights the security functions implemented by the TOE.

| Security functions | Features | Short description |
|---|---|---|
| **Endpoint protection** | Fine-grained control over process execution<br>Control over loaded libraries<br>File resources protection<br>Control over inbound and outbound network connections | The TOE is a solution designed specifically for protecting Satellite Control Centers.<br>The administrator can control which processes are executed, which libraries can be loaded in the system, the list of file resources that can be accessed and which inbound and outbound connections can be established by means of an ACL. This prevents the endpoint from executing unauthorized, unknown, or malicious applications |
| **Management** | Centralized management console<br>Identification and authentication of console users and role management | The TOE allows administrators and operators to configure, monitor, and manage the solution through a centralized web console. User access is restricted based on their role, identification and authentication are required to execute any action. |
| **Audit** | Audit review of ACL violations<br>Audit of user actions | Any breach of the ACL is audited as an alert in the TOE. These events can be monitored or queried in the central web console of the TOE. Any action performed by users within the management console is also audited. |

Table 2: Major security features

## 1.4. TOE DESCRIPTION

The TOE consists of three elements: the client, the server, which are the software *CosmicGuard for Linux endpoints v2.2*, and its accompanying documentation.

The TOE is comprised of these elements:

- The client consists of an agent that protects Linux endpoints by executing an ACL that prevents the execution of unwanted applications.
- The server consists of a centralized web management console for operating the solution and a platform that handles communications between the client and the server, called CosmicGuard Gateway, which is not a separate element but part of the server.
- The documentation needed for the installation [COSMICGUARD-AGD_PRE] and administration [COSMICGUARD-AGD_OPE] of the solution is provided in conjunction with the software.

### 1.4.1. PHYSICAL SCOPE OF THE TOE

The package that constitutes the TOE has the following cryptographic hash (SHA-256), which identifies it uniquely:

**94119DE95A2E16A55C9FFF84F514D510C457250976FBB46385DEB4C5EACAAA5D**

Inside the package, the following components will be found:

- **Host Management Server**
  - CosmicGuard Server v2.2 will be distributed as an EAR file
  - CosmicGuard Database scripts v2.2 will be distributed as an SQL script
  - CosmicGuard Gateway v2.2 will be distributed as a JAR file
- **Host Endpoint Agent**
  - CosmicGuard Agent v2.2 will be distributed as a single compressed file (tar.gz) containing all the necessary for the installation
- **Guidance**
  - CosmicGuard Installation Guide v2.2 [COSMICGUARD-AGD_PRE] will be distributed as a PDF file
  - CosmicGuard User Guide v2.2 [COSMICGUARD-AGD_OPE] will be distributed as a PDF file

*Note: recommended cipher suites are listed in the TOE installation guide* [COSMICGUARD-AGD_PRE].

#### 1.4.1.1. TOE delivery

The delivery procedure for the TOE is described below:

- GMV has a solution for distributing securely data with customers accessible through https://whisper.gmv.com , based on password.link, a software provided by e-Software. This solution is installed on-premises and offers a self-hosted file storage. The platform is administered, managed and maintained by GMV's IT department.

- Data is shared with the customer using the aforementioned solution. The share is done via a link which its access has read-only permission and it is protected with a password generated by the solution.
- The link is shared with the customer via email.
- The password is shared using a link in https://whisper.gmv.com which is publicly available. This service stores the password with no relationship to the shared link, but it can be configured to have an expiration date. And once the password is read it is then deleted, not allowing the next user to read it again.

#### 1.4.1.2. TOE architecture

The TOE's general architecture is illustrated in the following figure.

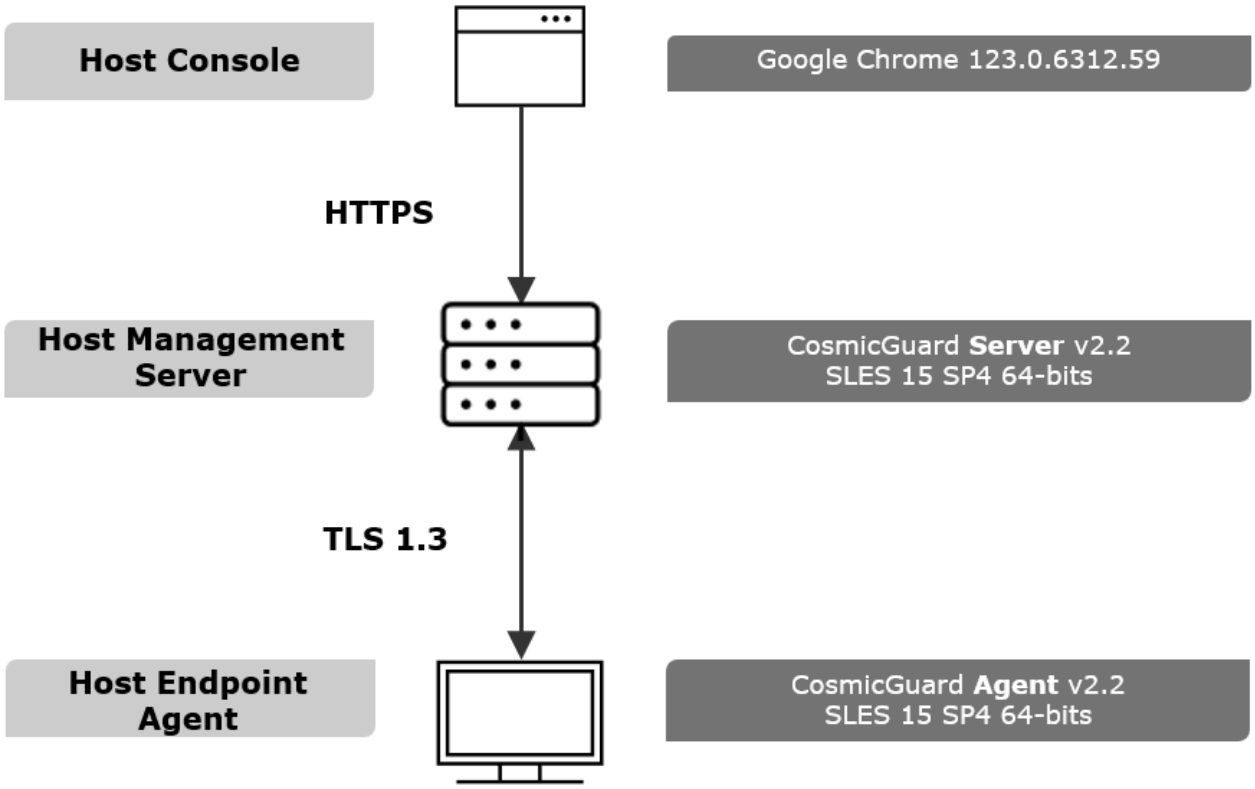


Figure 1: TOE's general architecture

The TOE can be deployed on virtual machines or dedicated hosts. GMV recommends deployment on virtual machines, which is the scenario that will be evaluated in this ST. In any case, the machines should be for the exclusive use of the TOE and should not be shared with other services. This is for security reasons and to isolate system administration tasks.

If we focus on the connections between the components of the system, it is important to emphasize the following:

- The communication between the Host Management and the Host Server must be HTTPS-based, with the use of certificates generated by the client's PKI infrastructure or by a recognized CA.
- Between the Host Agent and the Host Server, TLS v1.3 will be used, with mutual authentication enabled and the use of certificates generated by the client's PKI infrastructure or by a recognized CA.

In both cases, the recommended cipher suites are specified in the TOE Installation Guide [COSMICGUARD-AGD_PRE].

It is important to mention that an NTP server must be present and accessible from the endpoints and other machines within the client infrastructure. So that all the machines, whether physical or virtual, have the same time and date. There are no specific requirements regarding which version of the server to use or other features. In this case, it is recommended that the customer's policy be followed.

The TOE's detailed architecture is illustrated in the following figure.

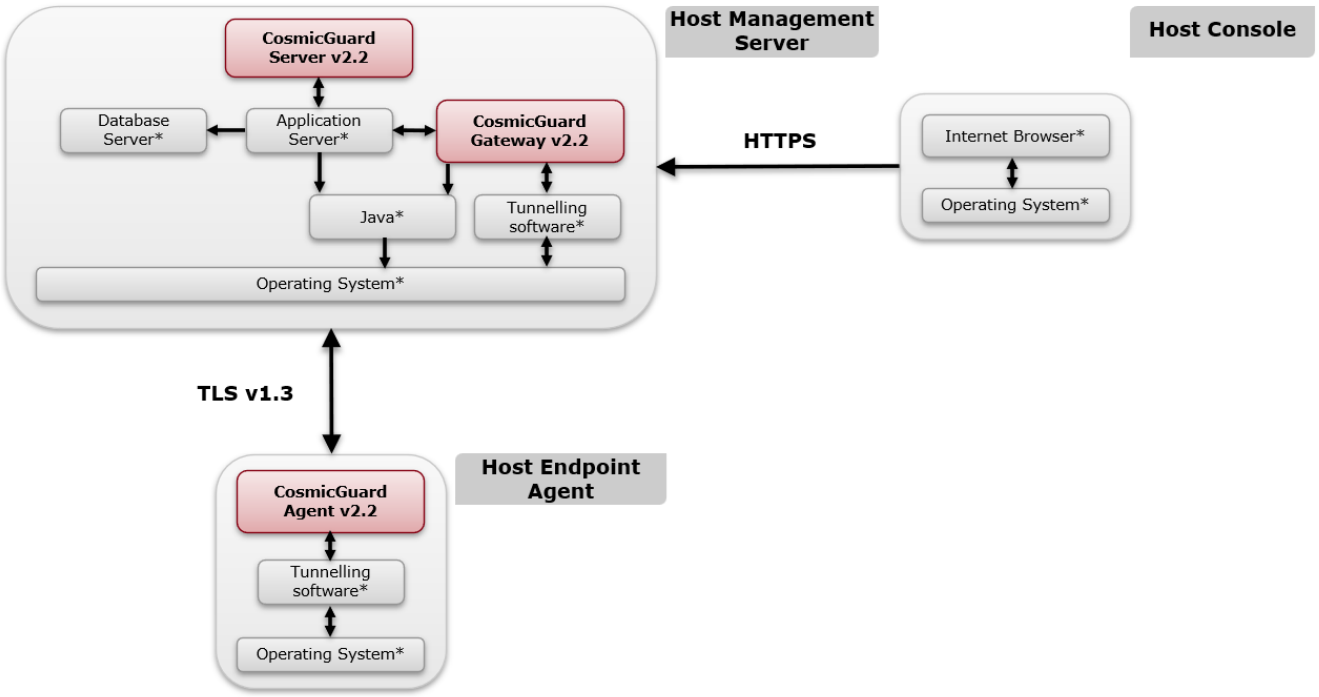


Figure 2: TOE's detailed architecture

*Please note that those elements marked with * are non-TOE components, and those in red are the TOE.*

#### 1.4.1.3. Non-TOE Components

The TOE requires the following components in the environment of evaluation:

- Host Console:
  - Any hardware and operating system that supports the following:
    - Internet browser, Google Chrome 123.0.6312.59 (64-bits) or greater.
- Host Management Server:
  - Any hardware and operating system that supports the following:
    - Database Server, PostgreSQL 15.5.
    - Application Server, WildFly 26.
    - Java, OpenJDK 11.
    - Tunnelling Software, GhosTunnel 1.7.2 supporting TLS v1.3.
- Host Endpoint Agent
  - Any hardware that supports the following:
    - Operating System, SLES 15 SP4.
    - Tunnelling Software, GhosTunnel 1.7.2 supporting TLS v1.3.
- All hosts can be executed in a virtual environment that meets the same requirements.
- Other services, such as NTP and PKI Infrastructure must be provided in the client's infrastructure.

### 1.4.2. LOGICAL SCOPE OF THE TOE

#### 1.4.2.1. Endpoint protection

The TOE offers the following features:

- Enforces the ACL in the endpoints, so that they are protected from executing unwanted or unknown applications. Authorization is granted based on defined rules:
  - Process execution.
    Restricts the execution of processes in the endpoint to those listed in the ACL.
  - Loaded libraries.
    Can restrict the loading of libraries in the endpoint to those listed in the ACL, if defined.
  - File resources protection.
    Restricts the access to resources in the endpoint to those listed in the ACL.
  - Inbound and outbound network connection.
    Restricts the establishment of inbound or outbound network connections to those listed in the ACL.
- Uses a mechanism of self-protection so that the ACL cannot be queried, nor modified, validating the integrity of the data.

#### 1.4.2.2. Management

The TOE Management Console offers the following functions:

- Provides access to the users through a centralized web console.
- Definition and management of the ACL.
- Requires users to be correctly identified and authenticated before executing any action.
- Handles the users and user roles from the management console.

#### 1.4.2.3. Audit

The TOE offers the following audit functions:

- Generates events when a violation of the ACL occurs.
- Provides audit review for the violations of the ACL.
- Audits any actions performed by the management console users.

### 1.4.3. EVALUATED CONFIGURATION

The evaluated configuration of the TOE is composed by the following elements:

- A virtual machine executed in Virtual Box 7.0.24 (recommended) with CosmicGuard Server v2.2 running on SLES 15 SP4 64-bits (Host Management Server).
- A virtual machine, executed in Virtual Box 7.0.24 (recommended) with CosmicGuard Agent for Linux v2.2 running on SLES 15 SP4 64-bits (Host Endpoint Agent).
- Access to the web console using Google Chrome 123.0.6312.59 (64-bits) or greater (Host Console) running on any compatible Operating System.
- All devices connected to the same LAN, remote access is not allowed.
- Virtual machines having the same time and date.
- Secure Boot must be enabled in the hypervisor used, recommended Virtual Box 7.0.16.
- Certificates must be issued by a trusted certification authority.
- The solution must be configured so that only Linux endpoints are managed, this can be achieved by following specific sections in the TOE User Guide [COSMICGUARD-AGD_OPE].

# 2. CONFORMANCE CLAIMS

## 2.1. CC CONFORMANCE CLAIM

This Security Target and the TOE claim conformance to part 2 and part 3 of CC Version 3.1, Revision 5:

- CC part 2 conformant (CCMB-2017-04-002)
- CC part 3 conformant (CCMB-2017-04-003)

## 2.2. PP CONFORMANCE CLAIM

This Security Target does not claim conformance to a Protection Profile.

## 2.3. PACKAGE CONFORMANCE CLAIM

This Security Target claims conformance to EAL2.

## 2.4. CONFORMANCE RATIONALE

The ST does not claim conformance to any Protection Profile. Therefore, the conformance claim rationale is not applicable.

# 3. SECURITY PROBLEM DEFINITION

## 3.1. THREATS

The TOE is designed to protect the following primary assets:

- Endpoint data
- Endpoint communications
- Endpoint software
- Endpoint attached devices and resources

It also protects the following supporting assets:

- Endpoint TSF data
- TSF configuration

The TOE addresses the following threats:

| Threat | Description |
|---|---|
| **T.UNAUTH_SW** | An attacker might introduce unauthorized software processes into the endpoint by any means, the attacker might execute the unauthorized software aiming to gain access to the legitimate processes or data contained in the endpoint. |
| **T.OS_LIBS** | An attacker might use legitimate processes present in the endpoint to load OS libraries and use them to access attached devices. |
| **T.UNAUTH_FILE** | An attacker might use a legitimate process in the endpoint to unauthorized access data or processes contained in the endpoint. |
| **T.NET_ACCESS** | A program hosted on the endpoint may gain unauthorized access to the attached network potentially leaking sensitive data. |
| **T.SW_VULN** | An attacker might exploit previously unknown vulnerabilities on the endpoint processes using those processes to gain unauthorized access to data, OS libraries, or the communications network. |
| **T.ACL_CORRUPT** | The ACL stored on the endpoint side might be accidentally corrupted or intentionally modified by an attacker, thus resulting in ineffective security mechanisms. |
| **T.UNAUTH_MGMT** | An attacker gains unauthorized access to the TOE management and modify the TSF configuration, resulting in ineffective security mechanisms. |
| **T.ATTK_UNNOT** | An attacker attempts an unauthorized action on the endpoint side and that attempt stays unnoticed, therefore preventing any further protective action from the system owner |

Table 3: Threats

## 3.2. ORGANIZATIONAL SECURITY POLICIES

The following table describes the OSPs relevant to the operation of the TOE:

| OSP | Description |
|---|---|
| **P.MGMT_AUDIT** | The authorized administrators of the TOE shall be held accountable for their actions within the management console of the TOE. |

Table 4: OSPs

## 3.3. ASSUMPTIONS

The following assumptions are made on the operational environment of the TOE:

| Assumption | Description |
|---|---|
| **A.INSTALL** | The TOE installation will be performed by trained personnel following the provided guidance. |
| **A.PHYSICAL_SEC** | All the components of the TOE are located in a physically secure environment where only authorized personnel are allowed to access. |
| **A.TOE_COMM_SEC** | The operational environment shall provide a trusted channel between physically separated elements of the TOE (the endpoint agent and the management console). |
| **A.MGMT_COMM_SEC** | The operational environment shall provide encrypted communication mechanisms for the TOE administrators accessing the TOE management console. |
| **A.MGMT_NET** | The TOE management console communication interfaces are in a secure network where only authorized personnel within the organization are allowed to access. |
| **A.DATABASE** | The operational environment shall provide a database that guarantees integrity and confidentiality for the data exchanged with the management console. |
| **A.OS_HARDENING** | The operational environment shall provide hardened OS according to the procedures of the organization. |
| **A.BOOT_INTEGRITY** | The operational environment shall provide mechanisms to verify the integrity of the TOE components installed in the endpoints during the system boot. |
| **A.TIMESTAMP** | The operational environment shall provide reliable timestamps for all the components of the TOE. |
| **A.HONEST_ADMIN** | Administrators of the TOE management console are trusted, non-hostile and will follow provided guidance. |

Table 5: Assumptions

# 4. SECURITY OBJECTIVES

## 4.1. SECURITY OBJECTIVES FOR THE TOE

The following security objectives are defined for the TOE:

| Objective | Description |
|---|---|
| **O.PROC_EXEC_CTRL** | The endpoint side of the TOE will provide the capability to authorize which processes can be executed on the endpoint. |
| **O.OS_LIBS_CRTL** | The endpoint side of TOE will provide the capability to control which process is authorized to access OS libraries based on each process. |
| **O.FILE_CRTL** | The endpoint side of the TOE will provide functionality to avoid that processes access any unauthorized file resource based on each process. |
| **O.NET_CRTL** | The endpoint side of TOE will provide the capability to avoid unauthorized network communications based on each process. |
| **O.ACL_INTEGRITY** | The endpoint side of TOE will verify the integrity of the ACL stored locally. |
| **O.ADMIN_AUTH** | The management console of the TOE will provide mechanisms to ensure that only authenticated users access the management functions of the TOE. |
| **O.MGMT** | The management console TOE will provide centralized management and roles to isolate administrative actions. |
| **O.MGMT_AUDIT_GEN** | The management console of the TOE will generate audit information any security-relevant actions taken by the users of the console. |
| **O.ACL_AUDIT_GEN** | The endpoint side of the TOE will generate an audit record for any ACL violations. |
| **O.ACL_AUDIT_REV** | The management console TOE will provide the capability to the authorized users to selectively review the ACL violations. |

Table 6: Security objectives for the TOE

## 4.2. SECURITY OBJECTIVES FOR THE ENVIRONMENT

The following security objectives are defined for the TOE operational environment:

| Objective | Description |
|---|---|
| **OE.ADMIN_GUIDANCE** | The TOE administrators will be provided with the necessary information for secure delivery, installation, configuration and management. |
| **OE.PHYSICAL_SEC** | All the components of the TOE are located in a physically secure environment where only authorized personnel are allowed to access. |
| **OE.TOE_COMM_SEC** | The operational environment shall provide a trusted channel between physically separated elements of the TOE. |
| **OE.MGMT_COMM_SEC** | The operational environment shall provide encrypted communication mechanisms for the TOE administrators accessing the TOE management console. |
| **OE.MGMT_NET** | The TOE management console communication interfaces are in a secure network where only authorized personnel within the organization are allowed to access. |
| **OE.DATABASE** | The operational environment shall provide a database that guarantees integrity and confidentiality for the data exchanged within the management console component. |
| **OE.OS_HARDENING** | The operational environment shall provide hardened OS according to the procedures of the organization. |
| **OE.BOOT_INTEGRITY** | The operational environment shall provide mechanisms to verify the integrity of the TOE components installed in the endpoints during the system boot. |
| **OE.TIMESTAMP** | The operational environment shall provide reliable timestamps for all the components of the TOE. |
| **OE.HONEST_ADMIN** | Administrators of the TOE management console are trusted, non-hostile and will follow provided guidance. |

Table 7: Security objectives for the TOE operational environment

## 4.3. SECURITY OBJECTIVES RATIONALE

### 4.3.1. SECURITY OBJECTIVES FOR THE TOE RATIONALE

The following table lists the security objectives for the TOE and shows which objectives are necessary to counter each threat or satisfy each or organizational security policies.

| THREAT OR OSP / OBJECTIVE FOR THE TOE | O.PROC_EXEC_CTRL | O.OS_LIBS_CRTL | O.FILE_CRTL | O.NET_CRTL | O.ACL_INTEGRITY | O.ADMIN_AUTH | O.MGMT | O.ACL_AUDIT_GEN | O.ACL_AUDIT_REV | O.MGMT_AUDIT_GEN |
|---|---|---|---|---|---|---|---|---|---|---|
| **T.UNAUTH_SW** | X | | | | | | | | | |
| **T.OS_LIBS** | | X | | | | | | | | |
| **T.UNAUTH_FILE** | | | X | | | | | | | |
| **T.NET_ACCESS** | | | | X | | | | | | |
| **T.SW_VULN** | X | X | X | X | | | | | | |
| **T.ACL_CORRUPT** | | | | | X | | | | | |
| **T.UNAUTH_MGMT** | | | | | | X | X | | | |
| **T.ATTK_UNNOT** | | | | | | | | X | X | |
| **P.MGMT_AUDIT** | | | | | | | | | | X |

Table 8: Security objectives mapping for threats and OSPs

The following table shows why the chosen objectives for the TOE are sufficient to counter a threat or satisfy an organizational security:

| THREAT OR OSP | OBJECTIVE | RATIONALE |
|---|---|---|
| **T.UNAUTH_SW** | **O.PROC_EXEC_CTRL** | Mitigates this threat by blocking any process that has not been previously included in the ACL. |
| **T.OS_LIBS** | **O.OS_LIBS_CRTL** | Mitigates this threat by blocking the use of operating system libraries to processes if not specified in the ACL. |
| **T.UNAUTH_FILE** | **O.FILE_CRTL** | Mitigates this threat blocking the access of file resource for each process if not previously specified in the ACL. |
| **T.NET_ACCESS** | **O.NET_CRTL** | Mitigates this threat blocking any network connection that has not been previously specified for a given process in the ACL. |
| **T.SW_VULN** | **O.PROC_EXEC_CTRL** | The exploitation of software vulnerabilities is mitigated by blocking a vulnerable process the execution of any new process or any other process that has not been defined in the ACL. |
| | **O.OS_LIBS_CRTL** | The exploitation of software vulnerabilities is mitigated by blocking the loading of operating system libraries by the vulnerable process. |
| | **O.FILE_CRTL** | The exploitation of software vulnerabilities is mitigated by blocking the access to any file resource that was not specified in the ACL |
| | **O.NET_CRTL** | The exploitation of software vulnerabilities is mitigated by blocking any network connection that was not specified in the ACL for that specific process. |
| **T.ACL_CORRUPT** | **O.ACL_INTEGRITY** | Mitigates this threat by validating the integrity of the stored ACL. |

| | | |
|---|---|---|
| **T.UNAUTH_MGMT** | **O.ADMIN_AUTH** | Mitigates this threat by providing authentication mechanisms, thus preventing unauthorized individuals to perform any operations on the TOE. |
| | **O.MGMT** | The TOE mitigates this threat by providing centralized and secured management |
| **T.ATTK_UNNOT** | **O.ACL_AUDIT_GEN** | The TOE mitigates this threat by generating audit information on the endpoint side for any action that violates the defined ACL. |
| | **O.ACL_AUDIT_REV** | The TOE mitigates this threat by providing the capability to review the audit registry generated for the ACL violations. |
| **P.MGMT_AUDIT** | **O.MGMT_AUDIT_GEN** | The TOE fulfills this OSP by generating audit information for any security-relevant actions performed by an administrator within the ACL. |

Table 9: Security objectives for the TOE rationale

## 4.3.2. SECURITY OBJECTIVES FOR THE TOE OPERATIONAL ENVIRONMENT RATIONALE

The following table lists the security objectives for the TOE operational environment and shows which objective are necessary to satisfy each assumption.

| ASSUMPTION \ OBJECTIVE FOR THE TOE OE | OE.ADMIN_GUIDANCE | OE.PHYSICAL_SEC | OE.TOE_COMM_SEC | OE.MGMT_COMM_SEC | OE.MGMT_NET | OE.DATABASE | OE.OS_HARDENING | OE.BOOT_INTEGRITY | OE.TIMESTAMP | OE.HONEST_ADMIN |
|---|---|---|---|---|---|---|---|---|---|---|
| **A.INSTALL** | X | | | | | | | | | |
| **A.PHYSICAL_SEC** | | X | | | | | | | | |
| **A.TOE_COMM_SEC** | | | X | | | | | | | |
| **A.MGMT_COMM_SEC** | | | | X | | | | | | |
| **A.MGMT_NET** | | | | | X | | | | | |
| **A.DATABASE** | | | | | | X | | | | |
| **A.OS_HARDENING** | | | | | | | X | | | |
| **A.BOOT_INTEGRITY** | | | | | | | | X | | |
| **A.TIMESTAMP** | | | | | | | | | X | |
| **A.HONEST_ADMIN** | | | | | | | | | | X |

Table 10: Security objectives mapping for threats and policies

The following table shows why the chosen objectives for the TOE operational environment are sufficient satisfy each assumption:

| ASSUMPTION | OBJECTIVE | RATIONALE |
|---|---|---|
| **A.INSTALL** | **OE.ADMIN_GUIDANCE** | Upholds the assumption by restating it as an objective for the operational environment. |
| **A.PHYSICAL_SEC** | **OE.PHYSICAL_SEC** | Upholds the assumption by restating it as an objective for the operational environment. |
| **A.TOE_COMM_SEC** | **OE.TOE_COMM_SEC** | Upholds the assumption by restating it as an objective for the operational environment. |
| **A.MGMT_COMM_SEC** | **OE.MGMT_COMM_SEC** | Upholds the assumption by restating it as an objective for the operational environment. |
| **A.MGMT_NET** | **OE.MGMT_NET** | Upholds the assumption by restating it as an objective for the operational environment. |
| **A.DATABASE** | **OE.DATABASE** | Upholds the assumption by restating it as an objective for the operational environment. |
| **A.OS_HARDENING** | **OE.OS_HARDENING** | Upholds the assumption by restating it as an objective for the operational environment. |
| **A.BOOT_INTEGRITY** | **OE.BOOT_INTEGRITY** | Upholds the assumption by restating it as an objective for the operational environment. |
| **A.TIMESTAMP** | **OE.TIMESTAMP** | Upholds the assumption by restating it as an objective for the operational environment. |
| **A.HONEST_ADMIN** | **OE.HONEST_ADMIN** | Upholds the assumption by restating it as an objective for the operational environment. |

Table 11: Security objectives for the TOE operational environment rationale

# 5. EXTENDED COMPONENTS DEFINITION

This ST does not define any extended component. All stated security requirements are present in CC Part 2 or in CC Part 3.

# 6. SECURITY REQUIREMENTS

This section defines the Security Requirements met by the TOE and the TOE environment. These requirements are contained in CC Part 2 and CC Part 3.

The CC allows four functional component operations (iteration, assignment, selection, and refinement) to be performed on functional requirements. This ST will represent each operation as follows:

- Iteration: identified by a numbered sequence (e.g. "ABC_CDE(1)").
- Assignment: enclosed in square brackets in italics (e.g. [*example assignment*]).
- Selection: enclosed in square brackets in regular text (e.g. [example selection]).
- Refinement: no refinement operations are performed on this ST.

## 6.1. SECURITY FUNCTIONAL REQUIREMENTS

The TOE's Security Functional Requirements (SFRs) addressed in this document are summarized in the following table:

| CLASS | SFR | NAME |
|---|---|---|
| **FAU** Security Audit | **FAU_GEN.1(1)** | Audit data generation (EVENTS) |
| | **FAU_GEN.1(2)** | Audit data generation (MGMT) |
| | **FAU_GEN.2** | User identity association |
| | **FAU_SAR.1** | Audit review |
| | **FAU_SAR.2** | Restricted audit review |
| **FDP** User Data Protection | **FDP_ACC.1(1)** | Subset access control (ENDPOINT) |
| | **FDP_ACC.1(2)** | Subset access control (MGMT) |
| | **FDP_ACF.1(1)** | Security attribute based access control (ENDPOINT) |
| | **FDP_ACF.1(2)** | Security attribute based access control (MGMT) |
| | **FDP_SDI.2** | Stored data integrity monitoring and action |
| **FIA** Identification and Authentication | **FIA_AFL.1** | Authentication failure handling |
| | **FIA_UAU.2** | User authentication before any action |
| | **FIA_UID.2** | User identification before any action |
| | **FIA_SOS.1** | Verification of secrets |

| | | |
|---|---|---|
| **FMT** Security Management | **FMT_MSA.1(1)** | Management of security attributes (ENDPOINT) |
| | **FMT_MSA.1(2)** | Management of security attributes (MGMT) |
| | **FMT_MSA.3(1)** | Static attribute initialization (ENDPOINT) |
| | **FMT_MSA.3(2)** | Static attribute initialization (MGMT) |
| | **FMT_SMF.1** | Specification of Management Functions |
| | **FMT_SMR.1** | Security roles |
| **FTA** TOE Access | **FTA_SSL.2** | User-initiated locking |
| | **FTA_SSL.3** | TSF-initiated termination |

Table 12: List of security functional requirements

## 6.1.1. CLASS FAU: SECURITY AUDIT

### 6.1.1.1. FAU_GEN.1(1): Audit data generation (EVENTS)

**FAU_GEN.1(1).1** **The TSF shall be able to generate an audit record of the following auditable events:**

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

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

**c) [*Events listed on Table 13*].**

**FAU_GEN.1(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, [*Additional information listed on Table 13*].**

| SFR | AUDITABLE EVENT | NOTES |
|---|---|---|
| **FDP_ACF.1(1)** | Violations of the ACL | |

Table 13: List of auditable events for FAU_GEN.1(1)

### 6.1.1.2. FAU_GEN.1(2): Audit data generation (MGMT)

**FAU_GEN.1(2).1** **The TSF shall be able to generate an audit record of the following auditable events:**

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

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

**c) [*Events listed on Table 14*].**

**FAU_GEN.1(2).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, [*Additional information listed on Table 14*].**

| SFR | AUDITABLE EVENT | NOTES |
|---|---|---|
| **FIA_UAU.2** | Successful and unsuccessful use of the authentication mechanism | It includes the user identity provided to the TOE |
| **FIA_UID.2** | Successful and unsuccessful use of the identification mechanism | It includes the user identity provided to the TOE |
| **FMT_SMF.1** | Object modifications | |

Table 14: List of auditable events for FAU_GEN.1(2)

### 6.1.1.3. 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 caused the event.**

### 6.1.1.4. FAU_SAR.1: Audit review

**FAU_SAR.1.1** **The TSF shall provide [*authorized users having the roles Event Administrator or Event Operator*] with the capability to read [*events listed on Table 13*] 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.**

#### 6.1.1.5. 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.**

***APPLICATION NOTE 1: No user is allowed to modify or delete any audit records.***

### 6.1.2. CLASS FDP: USER DATA PROTECTION

#### 6.1.2.1. FDP_ACC.1(1): Subset access control (ENDPOINT)

**FDP_ACC.1(1).1** **The TSF shall enforce the [*endpoint access control SFP*] on [*Subjects, objects and operations listed on Table 15*].**

| SUBJECT | OBJECT | OPERATION |
|---|---|---|
| PROCESSES | PROCESSES | EXECUTE |
| | LIBRARIES | LOAD |
| | FILES | EXECUTE<br>READ<br>READ_WRITE<br>RENAME<br>CHG_PERMISSION<br>SYMLINK<br>LINK<br>PIVOT_ROOT<br>CHROOT<br>MOUNT<br>UNMOUNT |
| | NETWORK CONNECTIONS | ESTABLISH NETWORK CONNECTIONS |

Table 15: Endpoint access control SFP

### 6.1.2.2. FDP_ACC.1(2): Subset access control (MGMT)

**FDP_ACC.1(2).1** **The TSF shall enforce the [*user access control SFP*] on [*Subjects, objects and operations listed on Table 16*].**

| SUBJECT | OBJECT | OPERATION |
|---|---|---|
| AUTHENTICATED USERS | MANAGEMENT SERVER OBJECTS | OPERATIONS BETWEEN SUBJECTS AND OBJECTS COVERED BY THE SFP, READ, WRITE |
| AUTHENTICATED USERS | USER OWN PASSWORD | MODIFY ITS OWN PASSWORD |
| ADMINISTRATOR USERS | USERS | READ, CREATE, DELETE, BLOCK, UNBLOCK AND MODIFY PASSORD |

Table 16: User access control SFP

### 6.1.2.3. FDP_ACF.1(1): Security attribute based access control (ENDPOINT)

**FDP_ACF.1(1).1** **The TSF shall enforce the [*endpoint access control SFP*] to objects based on the following: [*all operations between subjects and objects defined on Table 15*].**

**FDP_ACF.1(1).2** **The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*endpoint access control SFP must explicitly allow process execution, library use, files access, and network connections*].**

**FDP_ACF.1(1).3** **The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*rules, based on security attributes, that explicitly authorize access of subjects to objects as listed on Table 17*].**

**FDP_ACF.1(1).4** **The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [*no additional rules*].**

| OBJECT | SECURITY ATTRIBUTES |
|---|---|
| **PROCESSES** | PATH |
| **LIBRARIES** | ALLOWED/DISALLOWED |
| **FILES** | PATH |
| **NETWORK CONNECTIONS** | IP ADDRESS<br>PORT NUMBER<br>PROTOCOL (TCP/UDP)<br>DIRECTION (INBOUND/OUTBOUND) |

Table 17: Objects and security attributes for the endpoint access control SFP

### 6.1.2.4. FDP_ACF.1(2): Security attribute based access control (MGMT)

**FDP_ACF.1(2).1** **The TSF shall enforce the [*user access control SFP*] to objects based on the following: [*all operations between subjects and objects defined on Table 16*].**

**FDP_ACF.1(2).2** **The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*user access control SFP must explicitly allow object modifications within the management console*].**

**FDP_ACF.1(2).3** **The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*rules, based on security attributes, that explicitly authorize access of subjects to objects as listed on Table 18*].**

**FDP_ACF.1(2).4** **The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [*no additional rules*].**

| OBJECT | SECURITY ATTRIBUTES |
|---|---|
| **NETWORK MANAGEMENT** | NETWORK ADMINISTRATOR ROLES |
| **ACL MANAGEMENT** | ACL ADMINISTRATOR ROLES<br>ACL OPERATOR ROLES |
| **ACL VIOLATION EVENTS** | EVENT ADMINISTRATOR ROLES<br>EVENT OPERATOR ROLES |
| **USER MANAGEMENT** | USER ADMINISTRATOR ROLES<br>USER OPERATOR ROLES |

Table 18: Objects and security attributes for the user access control SFP

#### 6.1.2.5. FDP_SDI.2: Stored data integrity monitoring and action

**FDP_SDI.2.1** **The TSF shall monitor user data stored in containers controlled by the TSF for [*integrity errors on the endpoint access control SFP*] on all objects, based on the following attributes: [*integrity based on SHA256 hash algorithm*].**

**FDP_SDI.2.2** **Upon detection of a data integrity error, the TSF shall [*A restrictive default endpoint access control SFP is applied and an attempt to retrieve a valid SFP will be made afterwards*].**

### 6.1.3. CLASS FIA: IDENTIFICATION AND AUTHENTICATION

#### 6.1.3.1. FIA_AFL.1: Authentication failure handling

**FIA_AFL.1.1** **The TSF shall detect when [an administrator configurable positive integer within [*1 and 20*]] unsuccessful authentication attempts occur related to [*authentication attempts to the TOE administration console*].**

**FIA_AFL.1.2** **When the defined number of unsuccessful authentication attempts has been [met], the TSF shall [*forbid further authentication attempts to the TOE administration console*].**

#### 6.1.3.2. FIA_SOS.1: Verification of secrets

**FIA_SOS.1.1** **The TSF shall provide a mechanism to verify that secrets meet [*the following configurable requirements for user credentials:***

- ***Minimum username length***
- ***Minimum password length***
- ***Minimum number of capital letters in the password***
- ***Minimum number of lower-case letters in the password***
- ***Minimum number of digit letters in the password***
- ***Minimum number of special letters in the password*].**

***APPLICATION NOTE 1: the range for each configurable value is described below:***

- ***Minimum username length: 4-32 (defaults to 4)***
- ***Minimum password length: 4-99 (defaults to 6)***
- ***Minimum number of capital letters in the password: 0-99* (defaults to 0)***
- ***Minimum number of lower-case letters in the password: 0-99* (defaults to 0)***
- ***Minimum number of digit letters in the password: 0-99* (defaults to 0)***
- ***Minimum number of special letters in the password: 0-99* defaults to 0)***
- ***(*) The sum of capital, lower-case, digit and special letters in the password must be equal or lower than the minimum password length***

***APPLICATION NOTE 2: the following values are the minimum password strength and password management to prevent an attacker from guessing the password using brute force. Define the following values or higher (if required by your organization password policy):***

- ***Minimum username length: 8 (or higher)***
- ***Minimum password length: 12 (or higher)***
- ***Minimum number of capital letters: 1 (or higher)***
- ***Minimum number of lower-case letters: 1 (or higher)***
- ***Minimum number of digits: 1 (or higher)***
- ***Minimum number of special characters: 1 (or higher)***
- ***Maximum failed logins: 10 (or lower)***
- ***Expiration time: 90 (or lower)***
- ***Force password change: TRUE***
- ***Number of old passwords stored: 10 (or higher)***
- ***Allow non-expiring passwords: FALSE***

#### 6.1.3.3. 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.**

#### 6.1.3.4. 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.**

### 6.1.4. CLASS FMT: SECURITY MANAGEMENT

#### 6.1.4.1. FMT_MSA.1(1): Management of security attributes (ENDPOINT)

**FMT_MSA.1(1).1** **The TSF shall enforce the [*endpoint access control SFP*] to restrict the ability to [query, modify, delete] the security attributes [*list of security attributes included in Table 17*] to [*users with the authorized roles*].**

#### 6.1.4.2. FMT_MSA.1(2): Management of security attributes (MGMT)

**FMT_MSA.1(2).1** **The TSF shall enforce the [*user access control SFP*] to restrict the ability to [create, query, modify, delete] the security attributes [*list of security attributes included in Table 18*] to [*users with the authorized roles*].**

#### 6.1.4.3. FMT_MSA.3(1): Static attribute initialization (ENDPOINT)

**FMT_MSA.3(1).1** **The TSF shall enforce the [*endpoint access control SFP*] to provide [restrictive] default values for security attributes that are used to enforce the SFP.**

**FMT_MSA.3(1).2** **The TSF shall allow the [*ACL administrator*] to specify alternative initial values to override the default values when an object or information is created.**

#### 6.1.4.4. FMT_MSA.3(2): Static attribute initialization (MGMT)

**FMT_MSA.3(2).1** **The TSF shall enforce the [*user access control SFP*] to provide [restrictive] default values for security attributes that are used to enforce the SFP.**

**FMT_MSA.3(2).2** **The TSF shall allow the [*User administrator*] to specify alternative initial values to override the default values when an object or information is created.**

#### 6.1.4.5. FMT_SMF.1: Specification of Management Functions

**FMT_SMF.1.1** **The TSF shall be capable of performing the following management functions: [*manage the endpoint access control SFP,*

*manage the user access control SFP,*

*manage the authentication data for the management console,*

*automatically end user sessions after the configured maximum inactivity time,*

*and query the TOE version*].**

#### 6.1.4.6. FMT_SMR.1: Security roles

**FMT_SMR.1.1** **The TSF shall maintain the roles [**

***Event Administrator***
***Event Operator***
***Network Administrator***
***ACL Administrator***
***ACL Operator***
***User Administrator***
***User Operator*].**

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

## 6.1.5. CLASS FTA: TOE ACESS

### 6.1.5.1. FTA_SSL.2: User-initiated locking

**FTA_SSL.2.1** **The TSF shall allow user-initiated locking of the user's own interactive session, by:**

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

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

**FTA_SSL.2.2** **The TSF shall require the following events to occur prior to unlocking the session: [*user to re-authenticate*].**

### 6.1.5.2. FTA_SSL.3: TSF-initiated termination

**FTA_SSL.3.1** **The TSF shall terminate an interactive session after a [*configurable time of inactivity*].**

## 6.2. SECURITY FUNCTIONAL REQUIREMENTS RATIONALE

This section provides evidence supporting the internal consistency and completeness of the components in the Security Target.

The following table shows how requirements are mapped to the Security Objectives:

| SFR / OBJECTIVE | O.PROC_EXEC_CTRL | O.OS_LIBS_CRTL | O.FILE_CRTL | O.NET_CRTL | O.ACL_INTEGRITY | O.ADMIN_AUTH | O.MGMT | O.MGMT_AUDIT_GEN | O.ACL_AUDIT_GEN | O.ACL_AUDIT_REV |
|---|---|---|---|---|---|---|---|---|---|---|
| **FAU_GEN.1(1)** | | | | | | | | | X | |
| **FAU_GEN.1(2)** | | | | | | | | X | | |
| **FAU_GEN.2** | | | | | | | | X | | |
| **FAU_SAR.1** | | | | | | | | | | X |
| **FAU_SAR.2** | | | | | | | | | | X |
| **FDP_ACC.1(1)** | X | X | X | X | | | | | | |
| **FDP_ACC.1(2)** | | | | | | | X | | | |
| **FDP_ACF.1(1)** | X | X | X | X | | | | | | |
| **FDP_ACF.1(2)** | | | | | | | X | | | |
| **FDP_SDI.2** | | | | | X | | | | | |
| **FIA_AFL.1** | | | | | | X | | | | |
| **FIA_SOS.1** | | | | | | X | | | | |
| **FIA_UAU.2** | | | | | | X | | | | |
| **FIA_UID.2** | | | | | | X | | | | |
| **FMT_MSA.1(1)** | X | X | X | X | | | | | | |
| **FMT_MSA.1(2)** | | | | | | | X | | | |
| **FMT_MSA.3(1)** | X | X | X | X | | | | | | |
| **FMT_MSA.3(2)** | | | | | | | X | | | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **FMT_SMF.1** | | | | | | | X | | | |
| **FMT_SMR.1** | | | | | | | X | | | |
| **FTA_SSL.2** | | | | | | X | | | | |
| **FTA_SSL.3** | | | | | | X | | | | |

Table 19: Security functional requirements mapping to security objectives

The following table shows how each SFR meets each security objective:

| OBJECTIVE | SFR | RATIONALE |
|---|---|---|
| **O.PROC_EXEC_CTRL** | FDP_ACC.1(1) | Ensures audit data is generated on the endpoint side when an unauthorized process execution attempt is performed by a user or other process |
| | FDP_ACF.1(1) | Ensures that access to resources protected by the TOE is managed by rules |
| | FMT_MSA.1 (1) | Ensures the management of security attributes for processes on the endpoint |
| | FMT_MSA.3(1) | Ensures a default restrictive policy exists |
| **O.OS_LIBS_CRTL** | FDP_ACC.1(1) | Ensures that audit data is generated on the endpoint side when an authorized process attempts to load libraries, and that and the load permission is not granted to the process |
| | FDP_ACF.1(1) | Ensures that access to resources protected by the TOE is managed by rules |
| | FMT_MSA.1(1) | Ensures the management of security attributes for libraries on the endpoint |
| | FMT_MSA.3(1) | Ensures a default restrictive policy exists |
| **O.FILE_CRTL** | FDP_ACC.1(1) | Ensures audit data is generated on the endpoint side when an authorized process attempts to perform an operation on files that are not explicitly authorized |
| | FDP_ACF.1(1) | Ensures that access to resources protected by the TOE is managed by rules |
| | FMT_MSA.1(1) | Ensures the management of security attributes for file operations on the endpoint |
| | FMT_MSA.3(1) | Ensures a default restrictive policy exists |

| | | |
|---|---|---|
| **O.NET_CRTL** | FDP_ACC.1(1) | Ensures that audit data is generated on the endpoint side when an authorized process attempts to establish a network connection that is not explicitly authorized |
| | FDP_ACF.1(1) | Ensures that access to resources protected by the TOE is managed by rules |
| | FMT_MSA.1(1) | Ensures the management of security attributes for network connections on the endpoint |
| | FMT_MSA.3(1) | Ensures a default restrictive policy exists |
| **O.ACL_INTEGRITY** | FDP_SDI.2 | Ensures the ACL is not modified |
| **O.ADMIN_AUTH** | FIA_AFL.1 | Ensures that brute-force attacks on the authentication mechanisms are ineffective |
| | FIA_SOS.1 | Ensures that authentication secrets are complex enough and not easy to guess |
| | FIA_UAU.2 | Ensures users are authenticated before accessing the TOE data and administration |
| | FIA_UID.2 | Ensures that users are identified before accessing the TOE data and administration |
| **O.MGMT** | FDP_ACC.1(2) | Ensures that audit data is generated on the server side when users perform any security-related action on the TOE management console |
| | FDP_ACF.1(2) | Ensures that access to the TOE administration is managed by rules |
| | FMT_MSA.1(2) | Ensures the management of security attributes for users accessing the TOE management console |
| | FMT_MSA.3(2) | Ensures the initialization of security attributes for users accessing the TOE management console |
| | FMT_SMF.1 | Ensures audit data is generated for security-related actions on the TOE management console |
| | FMT_SMR.1 | Ensures that different actions on the TOE management console are divided in user roles |
| **O.MGMT_AUDIT_GEN** | FAU_GEN.1(2) | Ensures audit data is generated on the server side when users perform any security-related action on the management console |

| | | |
|---|---|---|
| | FAU_GEN.2 | Ensures that the user associated to each action is recorded to keep users accountable for their actions |
| **O.ACL_AUDIT_GEN** | FAU_GEN.1(1) | Ensures audit data is generated on the endpoint side for security-related events and any violations of the ACL |
| **O.ACL_AUDIT_REV** | FAU_SAR.1 | Ensures that audit data is provided by the TOE management console |
| | FAU_SAR.2 | Ensures that access to the audit data is restricted to users except those granted with the appropriate permissions |

Table 20: Security functional requirements rationale for security objectives

## 6.3. SECURITY FUNCTIONAL REQUIREMENTS DEPENDENCY RATIONALE

All dependencies of the SFRs (stated in CC part 2) are satisfied as described in the following table:

| SFR | HIERARCHICAL TO | DEPENDENCIES | SATISFIED BY |
|---|---|---|---|
| **FAU_GEN.1(1)** | none | FPT_STM.1 | OE.TIMESTAMP |
| **FAU_GEN.1(2)** | none | FPT_STM.1 | OE.TIMESTAMP |
| **FAU_GEN.2** | none | FAU_GEN.1<br>FIA_UID.1 | FAU_GEN.1(2)<br>FIA_UID.1 |
| **FAU_SAR.1** | none | FAU_GEN.1 | FAU_GEN.1 |
| **FAU_SAR.2** | none | FAU_SAR.1 | FAU_SAR.1 |
| **FDP_ACC.1(1)** | none | FDP_ACF.1 | FDP_ACF.1(1) |
| **FDP_ACC.1(2)** | none | FDP_ACF.1 | FDP_ACF.1(2) |
| **FDP_ACF.1(1)** | none | FDP_ACC.1<br>FMT_MSA.3 | FDP_ACC.1(1)<br>FMT_MSA.3(1) |
| **FDP_ACF.1(2)** | none | FDP_ACC.1<br>FMT_MSA.3 | FDP_ACC.1(2)<br>FMT_MSA.3(2) |
| **FDP_SDI.2** | none | none | - |
| **FIA_AFL.1** | none | FIA_UAU.1 | FIA_UAU.2 |
| **FIA_SOS.1** | none | none | - |
| **FIA_UAU.2** | FIA_UAU.1 | FIA_UID.1 | FIA_UID.2 |
| **FIA_UID.2** | FIA_UID.1 | none | - |
| **FMT_MSA.1(1)** | none | [FDP_ACC.1 or FDP_IFC.1]<br>FMT_SMR.1<br>FMT_SMF.1 | FDP_ACC.1(1)<br>FMT_SMR.1<br>FMT_SMF.1 |
| **FMT_MSA.1(2)** | none | [FDP_ACC.1 or FDP_IFC.1]<br>FMT_SMR.1<br>FMT_SMF.1 | FDP_ACC.1<br>FMT_SMR.1<br>FMT_SMF.1 |
| **FMT_MSA.3(1)** | none | FMT_MSA.1<br>FMT_SMR.1 | FMT_MSA.1(1)<br>FMT_SMR.1 |
| **FMT_MSA.3(2)** | none | FMT_MSA.1<br>FMT_SMR.1 | FMT_MSA.1(2)<br>FMT_SMR.1 |
| **FMT_SMF.1** | none | none | - |
| **FMT_SMR.1** | none | FIA_UID.1 | FIA_UID.2 |
| **FTA_SSL.2** | none | FIA_UAU.1 | FIA_UAU.2 |
| **FTA_SSL.3** | none | none | - |

Table 21: Security functional requirements dependency rationale

## 6.4. SECURITY ASSURANCE REQUIREMENTS

The TOE conforms to all security assurance requirements in EAL2 as defined in CC part 3. The following table lists all security assurance requirements (SARs):

| ASSURANCE CLASS | ASSURANCE COMPONENT |
|---|---|
| **ADV: Development** | ADV_ARC.1 Security architecture description |
| | ADV_FSP.2 Security-enforcing functional specification |
| | ADV_TDS.1 Basic design |
| **AGD: Guidance documents** | AGD_OPE.1 Operational user guidance |
| | AGD_PRE.1 Preparative procedures |
| **ALC: Life-cycle support** | ALC_CMC.2 Use of a CM system |
| | ALC_CMS.2 Parts of the TOE CM coverage |
| | ALC_DEL.1 Delivery procedures |
| **ASE: Security Target evaluation** | ASE_CCL.1 Conformance claims |
| | ASE_ECD.1 Extended components definition |
| | ASE_INT.1 ST introduction |
| | ASE_OBJ.2 Security objectives |
| | ASE_REQ.2 Derived security requirements |
| | ASE_SPD.1 Security problem definition |
| | ASE_TSS.1 TOE summary specification |
| **ATE: Tests** | ATE_COV.1 Evidence of coverage |
| | ATE_FUN.1 Functional testing |
| | ATE_IND.2 Independent testing - sample |
| **AVA: Vulnerability assessment** | AVA_VAN.2 Vulnerability analysis |

Table 22: Security assurance requirements for EAL2

## 6.5. SECURITY ASSURANCE REQUIREMENTS RATIONALE

EAL2 was chosen to provide a low to moderate level of assurance that is consistent with good commercial practices.

# 7. TOE SUMMARY SPECIFICATION

This section describes how the TOE meets each SFR. The following table summarizes how each SFR traces to each TOE security function.

| SFR | SFR NAME | TOE SECURITY FUNCTION |
|---|---|---|
| **FAU_GEN.1(1)** | Audit data generation (EVENTS) | AUDIT |
| **FAU_GEN.1(2)** | Audit data generation (MGMT) | AUDIT |
| **FAU_GEN.2** | User identity association | AUDIT |
| **FAU_SAR.1** | Audit review | AUDIT |
| **FAU_SAR.2** | Restricted audit review | AUDIT |
| **FDP_ACC.1(1)** | Subset access control (ENDPOINT) | ENDPOINT PROTECTION |
| **FDP_ACC.1(2)** | Subset access control (MGMT) | MANAGEMENT |
| **FDP_ACF.1(1)** | Security attribute based access control (ENDPOINT) | ENDPOINT PROTECTION |
| **FDP_ACF.1(2)** | Security attribute based access control (MGMT) | MANAGEMENT |
| **FDP_SDI.2** | Stored data integrity monitoring and action | ENDPOINT PROTECTION |
| **FIA_AFL.1** | Authentication failure handling | MANAGEMENT |
| **FIA_SOS.1** | Verification of secrets | MANAGEMENT |
| **FIA_UAU.2** | User authentication before any action | MANAGEMENT |
| **FIA_UID.2** | User identification before any action | MANAGEMENT |
| **FMT_MSA.1(1)** | Management of security attributes (ENDPOINT) | ENDPOINT PROTECTION |
| **FMT_MSA.1(2)** | Management of security attributes (MGMT) | MANAGEMENT |
| **FMT_MSA.3(1)** | Static attribute initialization (ENDPOINT) | ENDPOINT PROTECTION |
| **FMT_MSA.3(2)** | Static attribute initialization (MGMT) | MANAGEMENT |
| **FMT_SMF.1** | Specification of Management Functions | MANAGEMENT |
| **FMT_SMR.1** | Security roles | MANAGEMENT |
| **FTA_SSL.2** | User-initiated locking | MANAGEMENT |
| **FTA_SSL.3** | TSF-initiated termination | MANAGEMENT |

Table 23: SFR tracing to TOE security functions

## 7.1. AUDIT GENERATION

### 7.1.1. ENDPOINT EVENT GENERATION

Any ACL violation detected on the endpoint side is stored both locally in the endpoint and transmitted to the management server, and then stored on a database.

[FAU_GEN.1(1)]

### 7.1.2. ENDPOINT EVENT REVIEW

Any ACL violations are stored on the database and can be reviewed by authorized users with the appropriate roles using the management web console.

No user is allowed to modify or delete the audit records.

Storage duration can be configured in the console.

[FAU_SAR.1, FAU_SAR.2]

### 7.1.3. MANAGEMENT AUDIT

User actions performed within the management console are stored in a local log file and cannot be modified by any of the management console users.

Registered audit actions will be associated with the responsible user.

[FAU_GEN.1(2), FAU_GEN.2]

## 7.2. ENDPOINT PROTECTION

### 7.2.1. PROCESS EXECUTION

The ACL will define a whitelist of processes that can be launched on the endpoint.

Processes will be identified by their path.

[FDP_ACC.1(1), FDP_ACF.1(1)]

### 7.2.2. LIBRARY USE

The ACL will specify if a given authorized process (present in the process execution list) is allowed or not to use system libraries.

[FDP_ACC.1(1), FDP_ACF.1(1)]

### 7.2.3. FILE ACCESS

The ACL will specify which files or folders can be accessed by each authorized process, and what actions can be performed by the process.

The files or folders will be identified by their path.

The possible operations on a given file or folder are the following:

- EXECUTE

- READ
- READ_WRITE
- RENAME
- CHG_PERMISSION
- SYMLINK
- LINK
- PIVOT_ROOT
- CHROOT
- MOUNT
- UNMOUNT

[FDP_ACC.1(1), FDP_ACF.1(1)]

### 7.2.4. NETWORK CONNECTIONS

The security policy will specify a whitelist of network connections that can be stablished by each authorized process.

Connections will be identified by:

- IP address (either IPv4 or IPv6)
- Port
- Protocol (TCP or UDP)
- Direction (Inbound or outbound)

Also note that the rules can define ranges of IPs and ports.

[FDP_ACC.1(1), FDP_ACF.1(1)]

### 7.2.5. ACL INTEGRITY VALIDATION

The agent stores the SHA256 checksum of the file containing the ACL file defined by the administrator for each endpoint.

On every startup, the actual ACL file checksum is calculated and compared against the stored value.

If the calculated values are the same, the security functionality is started normally, otherwise, an attempt will be made to retrieve the correct security policy from the server.

[FDP_SDI.2]

## 7.3. MANAGEMENT

### 7.3.1. IDENTIFICATION AND AUTHENTICATION

Users accessing the management console will be identified by a username and a password. No action will be permitted until the users are fully authenticated.

Credentials for users of the management console are required to meet a minimum configurable strength (username and password length, number of capitals, lower case, digits and special characters)

Credentials will be stored in the local database. No clear text password will be stored; a hash derivation of the password will be stored using the Argon2id hash algorithm (salt length = 32B; hash length = 64B; parallelism = 1; memory = 19456B; iterations = 3).

Any login attempt (either successful or unsuccessful) will be registered.

Failed login attempts for known usernames will increase a counter of failed logins, if the configured limit of failed logins for a specific user is reached, the user is automatically locked. Users exceeding the maximum inactivity time will be automatically locked.

Locked users cannot access the management console and will require the intervention of a user administrator to unlock the account.

User sessions within the management console can be terminated by the user or will be automatically terminated after a configurable inactivity time.

[FIA_AFL.1, FIA_SOS.1, FIA_UAU.2, FIA_UID.2, FTA_SSL.2, FTA_SSL.3]

### 7.3.2. USER MANAGEMENT

The users can be created, modified, and deleted in the management console by an administrator user with the appropriate roles.

A user administrator role can also block and unblock users from entering the console and restore the user password. An active user can only modify its own password.

[FDP_ACC.1(2), FDP_ACF.1(2), FMT_MSA.1(2), FMT_MSA.3(2)]

### 7.3.3. USER ROLES AND PERMISSIONS

The user administrator (USER_ADMIN) can grant the roles and sub-roles to the users accessing the management console. The roles and all the possible sub-roles are listed in the following table:

| ROLES | SUB-ROLES | DESCRIPTION |
|---|---|---|
| **EVENT ADMINISTRATOR** | **EVENT_ADMIN** | A user with this role can perform any action related to the events. It contains every role in the Event Roles section |
| **EVENT OPERATOR** | **EVENT_OPERATOR** | A user with this role can search for events |
| | **EVENT_MONITOR_OPERATOR** | A user with this role can use the event monitor |
| | **EVENT_TREEMAP_OPERATOR** | A user with this role can see the tree-map tab in the events module |
| **NETWORK ADMINISTRATOR** | **NETWORK_ADMIN** | A user with this role can perform any action related to the network: it contains every other role in the Network admin section |
| | **NETWORK_ADMIN_TAB** | A user with this role has access to both Windows and Linux endpoint tab |
| | **NETWORK_EXEMODE_ELEMENT** | A user with this role can run the *change execution mode* command on an endpoint |
| | **NETWORK_EXEMODE_NODE** | A user with this role can run the *change execution mode* command on a node |
| | **NETWORK_ACL_ELEMENT** | A user with this role can run the *send ACL* command on an endpoint |
| | **NETWORK_ACL_NODE** | A user with this role can run the *send ACL* command on a node |
| | **NETWORK_REBOOT_ELEMENT** | A user with this role can run the *reboot/shutdown* command on an endpoint |

| | | |
|---|---|---|
| | **NETWORK_REBOOT_NODE** | A user with this role can run the *reboot/shutdown* command on a node |
| | **NETWORK_FLUSH_ELEMENT** | A user with this role can run the *flush events* command on an endpoint |
| | **NETWORK_FLUSH_NODE** | A user with this role can run the *flush events* command on a node |
| | **NETWORK_CONNECTION_OPERATOR** | A user with this role can see the values in the connection section |
| | **NETWORK_CONNECTION_ADMIN** | A user with this role can modify the values in the connection section |
| | **NETWORK_LOAD_FILE** | A user with this role can remotely obtain the logs from an endpoint |
| | **NETWORK_STRUCTURE_ADMIN** | A user with this role can modify the network structure by creating/deleting nodes and moving nodes or endpoints |
| | **NETWORK_STRUCTURE_MANTAINER** | A user with this role can modify the network structure by moving nodes or endpoints |
| | **NETWORK_RENAME_ELEMENT:** | A user with this role can rename an endpoint if the system is configured for server priority regarding endpoint names |
| | **NETWORK_INFORMATION_EDITOR** | A user with this role can modify the basic information of an endpoint, like the comments, model, manufacturer and other free input fields that have elements and nodes |
| | **NETWORK_DELETE_ELEMENT** | A user with this role can delete an endpoint, sending it to the ABANDONED node |
| | **NETWORK_INFORMATION_EDITOR** | A user with this role can modify the basic information of an endpoint, like the comments, model, manufacturer and other free input fields that have elements and nodes |
| | **NETWORK_DELETE_ELEMENT** | A user with this role can delete an endpoint, sending it to the ABANDONED node |
| | **NETWORK_DROP_ELEMENT** | A user with this role can permanently delete an endpoint from the ABANDONED node |
| | **NETWORK_DROP_ALL** | A user with this role can permanently delete all endpoints from the ABANDONED node |
| | **NETWORK_LOCAL_EVENT_RULES** | A user with this role can manually send the local event rules to a node or endpoint |
| **ACL ADMINISTRATOR** | **ACL_ADMIN** | A user with this role can perform any action related to the ACLs. It contains every other role in the ACL Roles section |
| | **ACL_EDITOR** | A user with this role can edit the values of the ACL (create/modify/delete rules) |
| | **ACL_VERSIONER** | A user with this role can move the ACL through its lifecycle |

| | | |
|---|---|---|
| **ACL OPERATOR** | **ACL_OPERATOR** | A user with this role has access to see the values of the ACL |
| | **ACL_DEPLOYMENT_OPERATOR** | A user with this role can access to the ACL deployment section and check if the endpoints have the ACL synchronized |
| **USER ADMINISTRATOR** | **USER_ADMIN** | A user with this role can modify/create/delete other users of the management console |
| | **USER_DISABLE_ACCOUNT** | A user with this role can disable an account (the disabled user won't be able to log into the console) |
| | **USER_DISABLE_DELETE_ACCOUNT** | A user with this role can disable an account (the disabled user won't be able to log into the console) and to permanently delete an account |
| | **USER_ENABLE_ACCOUNT** | A user with this role can enable a disabled account (the enabled user will be able to log into the console) |
| | **USER_CREATE** | A user with this role can create other users, change their name, as well as modify all its data, role permissions and network permissions |
| | **USER_EDIT** | A user with this role can modify the user information in the "data" tab, except for the user name |
| | **USER_ASSIGN_ROLES** | A user with this role can modify the role permission for other users (except USER_ADMIN, that can't be assigned except by other USER_ADMIN) |
| | **USER_ASSIGN_NETWORK** | A user with this role can define the network permissions (nodes visibility) for other users |
| | **USER_ASSIGN_GROUPS** | A user with this role can define which groups the user will be a member of |
| | **USER_GROUP_EDITOR** | A user with this role can create and edit groups |
| **USER OPERATOR** | **USER_GROUP_OPERATOR** | A user with this role can see the existing groups |
| | **USER_OPERATOR** | A user with this role can see the Users section and inspect all the user permissions, but it won't be allowed to perform any changes |

Table 24: User roles description

[FMT_SMF.1, FMT_SMR.1]

## 7.3.4. ACL MANAGEMENT

An ACL contains a list of rules, specifying what processes can be executed or which resources can be accessed when enforced in the endpoint machine.

**ACL Life cycle**

An ACL can have the following statuses during its life cycle:

- Learning: the ACL can be edited, but it can NOT be sent to the endpoints
- Versioned: the ACL can NOT be edited,
- Obsolete: the ACL can NOT be edited NOR sent to the endpoint

New ACLs can be created by an administrator with the appropriate roles, and they will be created in "Learning" status.

ACL status can only be changed by an administrator with the appropriate roles, and it can only be changed to the next status as per listed.

The "Obsolete" status is the final status and does not allow any change.

**ACL Content**

The ACL is structured by processes. A process must be explicitly included in the ACL to be executed on the endpoint machine.

Each process has a list of file resources and connections, they must be explicitly included in the ACL to be accessed by each process.

Each process has a flag for the loading of O.S. libraries, the flag must be enabled for each process to load O.S. libraries.

All the ACL data is stored by the management console on the database.

**ACL Assignment**

ACL in the "Versioned" status can be assigned to endpoints by administrators with the appropriate role.

ACL data is sent as a file to the endpoint side of the TOE.

[FMT_MSA.1(1), FMT_MSA.3(1)]

## 7.3.5. TOE VERSION QUERY

Fully authenticated users of the management console are able to query the TOE version number.

[FMT_SMF.1]

# 8. DOCUMENT REFERENCES

The following documents are considered to be applicable, being their versions and date the valid ones at this document publishing time:

| Code | Document |
|---|---|
| **CC part 1** | Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2017-04-001, Version 3.1, Revision 5, April 2017 |
| **CC part 2** | Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components; CCMB-2017-04-002, Version 3.1, Revision 5, April 2017 |
| **CC part 3** | Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components; CCMB-2017-04-003, Version 3.1, Revision 5, April 2017 |
| **CEM** | Common Methodology for Information Technology Security Evaluation, CCMB-2017-04-004, Version 3.1, Revision 5, April 2017 |
| **COSMICGUARD-AGD_PRE** | CosmicGuard 2.2 Installation Guide, Version 6 |
| **COSMICGUARD-AGD_OPE** | CosmicGuard 2.2 User Guide, Version 7 |

Table 25: Document references

# 9. DEFINITIONS AND ACRONYMS

## 9.1. DEFINITIONS

Concepts and terms used in this document needing a definition are included in the following table:

| Concept/Term | Definition |
|---|---|
| Administrator | Entity that has a level of trust with respect to all policies implemented by the TSF |
| Role | Pre-defined set of rules establishing the allowed interactions between a user and the TOE |
| Security Objective | Statement of an intent to counter identified threats and/or satisfy identified organizational security policies and/or assumptions. |
| Security Problem Definition | Statement, which in a formal manner defines the nature and scope of the security that the TOE is intended to address. |
| Security Requirement | Requirement, which is part of a TOE security specification as defined in a specific security target (ST) or in a protection profile (PP) |
| Security Target (ST) | Implementation-dependent statement of security requirements for a TO) based on a security problem definition |
| Target Of Evaluation (TOE) | Set of software, firmware and/or hardware accompanied by guidance, which is the subject of an evaluation |
| Threat Agent | Entity that has potential to exercise adverse actions on assets protected by the TOE |
| TOE security functionality (TSF) | Combined functionality of all hardware, software, and firmware of a TOE that is relied upon for the correct enforcement of the SFRs |
| Administrator | Entity that has a level of trust with respect to all policies implemented by the TSF |
| Role | Pre-defined set of rules establishing the allowed interactions between a user and the TOE |
| Security Objective | Statement of an intent to counter identified threats and/or satisfy identified organizational security policies and/or assumptions. |
| Security Problem Definition (SPD) | Statement, which in a formal manner defines the nature and scope of the security that the TOE is intended to address. |
| Security Requirement | Requirement, which is part of a TOE security specification as defined in a specific security target (ST) or in a protection profile (PP) |
| Security Target (ST) | Implementation-dependent statement of security requirements for a TO) based on a security problem definition |
| Target Of Evaluation (TOE) | Set of software, firmware and/or hardware possibly accompanied by guidance, which is the subject of an evaluation |
| Threat Agent | Entity that has potential to exercise adverse actions on assets protected by the TOE |
| TOE security functionality (TSF) | Combined functionality of all hardware, software, and firmware of a TOE that is relied upon for the correct enforcement of the SFRs |

Table 26: Definitions

## 9.2. ACRONYMS

Acronyms used in this document and needing a definition are included in the following table:

| Acronym | Definition |
|---|---|
| ACL | Access Control List – A list of permissions on objects |
| CA | Certificate Authority – An entity that issues digital certificates to verify the identity of organizations and individuals for secure communication. |
| CC | Common Criteria - An international standard (ISO/IEC 15408) for evaluating the security of IT products |
| EAL | Evaluation Assurance Level - A numerical grade assigned to an IT product or system after a security evaluation indicating the depth and rigor of the evaluation |
| EAR | Enterprise ARchive – A packaging format used to bundle multiple modules of a Java EE into a single file for distribution and deployment |
| HTTP(S) | Hypertext Transfer Protocol (Secure) – The protocol used for transmitting hypertext requests and information on a data network (Secure when the communication is encrypted) |
| IP | Internet Protocol – A communication protocol over a data network |
| IT | Information Technology |
| JAR | Java ARchive – A file format used to aggregate many Java class files and associated metadata and resources into a single file for distribution |
| LAN | Local Area Network – A network that connects computers and devices within a limited area |
| NTP | Network Time Protocol – A protocol used to synchronize the clocks of computers over a data network |
| OS | Operating System – The core software that controls a computer's hardware and software resources, enabling other programs to run and providing a user interface |
| PDF | Portable Document Format - A file format developed by Adobe that preserves the layout, formatting, and fonts of documents across different devices and platforms |
| PKI | Public Key Infrastructure – A framework for managing digital certificates and public-key encryption |
| PP | Protection Profile - A document that specifies an implementation-independent set of security requirements for a category of products |
| SAR | Security Assurance Requirements - Criteria used to evaluate the confidence in the security properties of a TOE |
| SLES | SuSE Linux Enterprise Server – A Linux-based operating system developed by SUSE for servers |
| SFR | Security Functional Requirements - Specific security requirements that a TOE must meet |
| SFP | Security Function Policy - Rules and guidelines that govern the behavior of the TOE in enforcing security |
| ST | Security Target – Document that identifies the security properties of a target of evaluation (TOE) and the security requirements it meets |
| TCP | Transmission Control Protocol – A core protocol of the Internet Protocol Suite that ensures reliable, ordered, and error-checked delivery of data between applications over a data network |
| TLS | Transport Layer Security – A protocol that ensures secure communication over a data network by encrypting the data exchanged between applications and their users |
| TOE | Target Of Evaluation – An IT product or system that is the subject of a security evaluation |
| TSF | TOE Security Functions - A set of all TOE components that must be relied upon for the correct enforcement of the SFRs |
| UDP | User Datagram Protocol - A communication protocol used for transmitting data over a network. Unlike TCP, UDP is connectionless and does not guarantee reliable delivery. |

Table 27: Acronyms