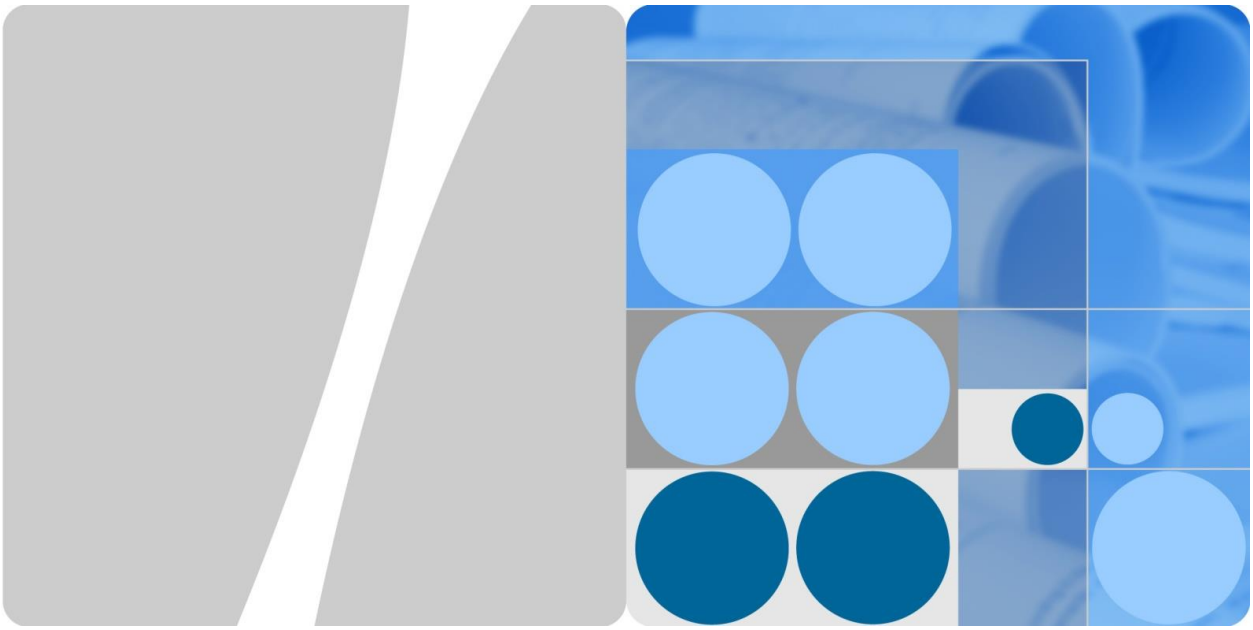


CC Huawei Reliable Telecomm Operating System Version Extra Exploration 301.1

Security Target

Version	6.0
Status	Released
Last update	2023-12-14
Classification	Public

HUAWEI TECHNOLOGIES CO., LTD.

Trademarks and Permissions

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

About This Document

Purpose

This document provides description about ST (Security Target) against BSI PP.

Change History

Changes between document issues are cumulative. The latest document issue contains all the changes made in earlier issues.

Date	Revision Version	Section Number	Change Description	Author
2021.07.05	1.0	All	Initial Draft	Yulu
2022.01.06	1.5	3.2	Modified on threat adverse action consequences.	Wanglei
2023.04.24	2.0	5 3.4 7.1	Added A.INTEGRATION_BOARD and OE.INTEGRATION_BOARD	Yulu, Wanglei
2023.07.29	2.5	1.2 1.4.2.1	Updated TOE version to 301.1.3	Yulu
2023.12.04	3.0	All	Delete RNG related SFRs and descriptions	Yulu, Wanglei
2023.12.13	5.0	1.4.2.1	Modify versions of AGD Release ST	Yulu
2023.12.14	6.0	1.4.2.1	Modify versions of AGD	Yulu

Contents

Index of Tables

1. Security Target Introduction

1.1 Security Target Reference

Name	CC Huawei Reliable Telecomm Operating System Version Extra Exploration 301.1 Security Target
Version	6.0
Date	2023-12-14
Sponsor	Huawei Technologies Co., Ltd
Developer	Huawei Technologies Co., Ltd

1.2 TOE Reference

Name	Huawei Reliable Telecomm Operating System Version Extra Exploration
Version	301.1.3
Developer	Huawei Technologies Co., Ltd

Huawei Reliable Telecomm Operating System Version Extra Exploration (RTOS V2X) contains a self-developed microkernel, along with user programs. The self-developed microkernel based operating system is facing challenges and opportunities in the industry. For example, microkernel is not widely used in the industry. Therefore, telecom communication application scenarios need further exploration, thus, it is named Version Extra Exploration (abbreviated with V2X, since there are two Xs inside the name).

1.3 TOE Overview

1.3.1 Configurations defined with this ST

This security target documents the security characteristics of the Huawei Reliable Telecomm Operating System Version Extra Exploration product (abbreviated with RTOS V2X throughout this document).

This Security Target refers to the following Protection profiles:

- [OSPP] Operating System Protection Profile, BSI-CC-PP-0067, Version 2.0; https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/PP/aktuell/PP_0067.html
- [OSPP-AM] OSPP Extended Package - Advanced Management, BSI-CC-PP-0067, OSPP EP-AM, Version 2.0; https://www.bsi.bund.de/SharedDocs/Zertifikate_CC/PP/aktuell/PP_0067.html

1.3.2 TOE Type

Huawei Reliable Telecomm Operating System Version Extra Exploration (RTOS V2X)is a highly-configurable self-developed microkernel-architecture operating system for embedded device.

Huawei RTOS V2X is an operating system that is composed by self-developed HongMeng Kernel and userspace packages. Please refer to 1.4.2.2 for the structure and operational environment illustration of the TOE.

1.3.2.1 Hardware Specifics

All security functions claimed in this ST apply to the following architectures, and these are not part

of the TOE:

- Physical devices compatible with the arm64 instruction set;

1.3.3 Non-TOE Hardware, Software, Firmware Supported

Non-TOE Hardware Identification: The following physical hardware platforms, corresponding firmware, and components are supported by the TOE, and these hardware are not part of the TOE:

①SD6221 SoC based hardware device board: The device board evaluated with the TOE is AAU 5636
With minimum requirements of:

- RAM: 350M
- Flash: 190M
- A serial port and a network port interface

The TOE is evaluated only on SD6221 SoC based hardware device board. In this evaluation, the hardware board with SD6221 SoC is AAU5636, along with a finished board to aid upload binary images of the TOE. The listed board contains physical peripheral devices (flash storage, network interface cards, serial interfaces, etc), and can be used with the TOE without affecting its security functions. Note that AAU is not the only device board that can be mounted. Device boards, using the listed SoCs, can also be mounted.

②Integration board: The integration board is used to check the signature of the TOE binary images to ensure secure initialization. It also deployed FTP server to transmitting images and NTP server to provide the right time, It is part of the operational environment.

③The finished board is used to aid upload binary images of the TOE, and it functions as a network port. The listed table is an example of the hardware specification of the finished board, the customer should choose and adapt the finished board based on the actual environment.

03025VQG	PHY-2.5V/3.3V-10BASE-T/100BASE-TX/FX-MII/RMII-TQFP64-1
03024AEQ	PHY-2.5V/3.3V-10BASE-T/100BASE-TX/FX-MII/RMII-TQFP64-1

Note that the intended use is targeting internal customers for integration purposes and the OE configuration of the finished boards are only accessible on Huawei platform.

④BIOS: BIOS served as the firmware for the TOE, used to provide runtime services for TOE and to perform hardware initialization during the booting process. The version of BIOS used in this evaluation is iWare WL V100R022C10SPC003B002, and it is not part of the TOE. Note that U-Boot could be used instead of BIOS in the evaluated scenario.

Note that the Hardware, integration board, AAU with SD6221 SoC board, along with the finished board, are not part of the TOE. The finished boards can be searched through Huawei platform. Potential firmware, such as BIOS, is a layer between the TOE and the hardware, and it is also without the TOE scope.

1.3.4 Intended Method of Use

1.3.4.1 Integration-purpose computing environment

The TOE is a self-developed multi-user multi-tasking microkernel-architecture operating system for embedded devices. It has the capability to provide services to multiple users, local or remote, at the same time. After a successful login, users gain access to a general computing environment which allows launching user applications, issuing user commands at shell level, creating and accessing files.

The TOE provides adequate mechanisms to separate users and protect their data. Privileged commands are restricted to administrative users only.

The TOE is intended to operate in a networked environment with other instantiations of the TOE as well as other well-behaved peer systems.

It is assumed that the responsibility for safeguarding user data protected by the TOE can be delegated to human users of the TOE if such users are allowed to log on and spawn processes on their own behalf. All user data is controlled by the TOE. The user data is stored in named objects, and the TOE can associate a description of the access rights to that object with each named object.

The TOE enforces controls such that access to data objects is in accordance with the access restrictions placed on that object by its owner and administrative users. Ownership of named objects can be transferred under the access control policies implemented by the TOE.

The TOE enforces discretionary access control policy, in which, access rights (e.g. read, write, execute) can be assigned to data objects with respect to subjects identified with their UID, GID and supplementary GIDs. Once a subject is granted access to an object, the content of that object can be used freely by the subject to influence other objects accessible to the same subject.

1.3.4.2 Operating Environment

The TOE permits multiple applications assigned with different UIDs to perform on a platform with one or more processors, peripheral devices and storage devices so as to execute functionalities requiring controlled shared access on system data. With different UIDs, proper access restrictions to resources assigned to processes can be enforced using the provided access control mechanism. Such usage scenarios are typical for systems accessed by processes, or by local users of the computer system, or by other users with protected access to the system.

1.3.5 Major Security Features

RTOS V2X is a multi-user, multi-tasking self-developed microkernel-architecture operating system. It provides a platform for a variety of applications.

RTOS V2X provides the following key security features:

- Security Audit: The TOE is able to intercept all system calls and record events occurred in the system. The security audit functionality also allows configuring the events to be audited, reviewing and searching the audit log retrieved.

- Cryptographic support: The TOE provides cryptographically secured communication to allow remote entities to log into the TOE using SSHv2 protocol.

- Identification and Authentication: The TOE includes several ways to identify and authenticate users (via local console using username and password or via SSH using password and public-key based authentication). The TOE also offers a password quality enforcement mechanism as well as authentication attempts control mechanism. The TOE is able to end user sessions after an inactive period of time. When the TOE re-access is asked, identification and authentication of users are required.

- User Data Protection: The TOE offers a Discretionary Access Control (DAC), which allows owner of named objects to control the access permissions of these objects. Moreover, the TOE kernel implements the *IPTables* mechanism in order to provide a packet filter at network and

transfer layer. Using these two mechanisms the TOE offers an access control policy as well as an information flow control policy.

- Security Management: The TOE offers to users and/or authorized administrators the capability of modifying the configuration of the TSF. The TOE allows local and remote management using OpenSSH.

- Trusted Channel: Using the cryptographic communication protocols mentioned above (SSH), the TOE is able to establish a secure and trusted communication channel with other IT entities.

These primary security features are supported by domain separation and reference mediation, which can ensure that the security features are always invoked and cannot be bypassed.

1.4 TOE Description

1.4.1 Introduction

RTOS V2X is a multi-user, multi-tasking self-developed microkernel-architecture operating system. It provides a platform for a variety of applications.

RTOS V2X evaluation covers a distributed network of systems running the evaluated version and its configurations as well as other peer systems operating within the same management domain. The hardware platforms selected for the evaluation consist of machines that are available on market.

The TOE Security Functions (TSFs) consist of functions of RTOS V2X that run in kernel mode and some trusted processes running in user mode. These are the functions that enforce the security policy defined in this Security Target. Tools and commands executed in user mode that are used by an administrative user also need to be trusted to manage the system in a secure way, but they are not considered to be part of this TSF, just as with other operating system evaluations.

The hardware, BIOS firmware and potentially other firmware layers between the hardware and the TOE, are considered to be part of the operational environment, but do not belong to the TOE scope. BIOS is used to process hardware and operating systems on starting up, and it is also between the hardware and the TOE, thus, it does not belong to the TOE scope.

The TOE includes standard networking applications, such as *sshd*(8), which allow to access the TOE via cryptographically protected communication channel.

System administration tools include standard command line tools. A graphical user interface for system administration or any other operations are not included in the evaluated configuration.

1.4.2 TOE boundaries

1.4.2.1 Physical boundaries

The OS binaries are all distributed via the Huawei Network. Huawei Network is an internal network and only can be accessed by Huawei employees.

The following OS binaries are provide for the TOE:

- RTOS_V2X_301.1.3.rootfs

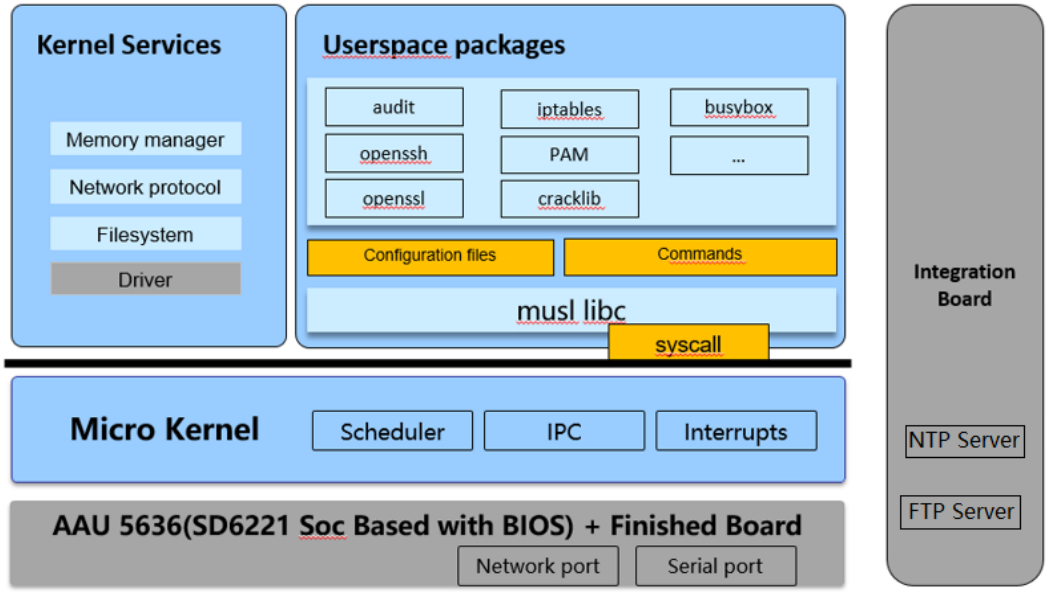
- RTOS_V2X_301.1.3.kernel
- RTOS_V2X_301.1.3.dtb

The following documentations are packed with the TOE:

- Installation guide: CC Huawei Reliable Telecomm Operating System Version Extra Exploration 301.1 AGD_PRE-v2.1, doc format.
- User guide: CC Huawei Reliable Telecomm Operating System Version Extra Exploration 301.1 AGD_OPE-v1.7, doc format.
- Manpage guide: CC Huawei Reliable Telecomm Operating System Version Extra 301.1 ALC_CMS_manpage-v0.3, tar package.

The list of hardware applicable to the TOE is given in Section 1.3.3.

1.4.2.2 Logical boundaries



The figure is the structure and operation environment of RTOS V2X, the grey boxes, driver and hardware device boards, are not part of the TOE. The TOE relies on an integration board to provide initial secure initialization of the binary images, and the integration board, which is also in the grey box, is not part of the TOE.

All security functions claimed in this ST apply to all architectures and systems allowed via this ST.

The primary security features of the TOE include:

- Cryptographic Communication: The TOE provides cryptographically secured communication to allow remote entities log into the TOE. The SSHv2 protocol is provided to set up interactive session with the TOE. The TOE provides both the server side and the client side applications. Using the OpenSSH suite, password-based and public-key-based authentication are allowed.

- Packet Filter: The TOE kernel implements layering structure of network protocols. It has IPTables mechanism to provide a stateful packet filter at network layer and transfer layer for regular IP-based communication.

- Identification and Authentication: User identification and authentication in the TOE includes all forms of interactive login (e.g. using the SSH protocol or log in at the local console) as well as identity changes through the command like su. These all rely on explicit authentication information provided interactively by a user.

  The authentication security function allows password-based authentication. For SSH access, public-key-based authentication is also supported.

  Password quality enforcement mechanisms offered by the TOE are enforced at the time when the password is changed.

- Discretionary Access Control (DAC): DAC allows owners of named objects to control the access permissions to these objects. The owners can permit or deny access by other users based on the configured permission settings. The DAC mechanism is also used to ensure that untrusted users cannot tamper with the TOE mechanisms.

- Auditing: The audit function is designed to be an audit system making RTOS V2X compliant with the requirements from OSPP. The audit system is able to intercept all system calls as well as retrieving audit log entries from privileged user space applications. The subsystem allows to configure the events to be actually audited from the set of all events that are possible to be audited, and to review and search audit logs retrieved.

- Security Management: The security management facilities provided by the TOE are available to authorized users and/or authorized administrators to modify the configuration of TSF. The TOE allows local management on local consoles and remote management via OpenSSH. Administrative users can log in remotely and perform the same management tasks as a locally operating administrator.

1.4.3 Evaluated configuration

The evaluated configuration is defined as follows:

- The package set evaluated by CC for the TOE must be selected at install time according to the installation guide and be installed accordingly.

- The TOE supports the use of IPv4, which is also supported in the evaluated configuration.

- The default configuration for identification and authentication include both the defined password-based PAM modules and the key-based authentication for OpenSSH. Support for other authentication options, e.g. smart card authentication, is not included in the evaluation configuration.

- If the system console is used, for example serial port connection, it must be connected directly to the TOE and afforded the same physical protection as the TOE.

Configurations and settings that are different from that specified in the installation guide are not permitted.

Moreover, the TOE was tested on the following physical platforms:

- SD6221 SoC based hardware device board (355M RAM, 192M flash storage, serial port and network port interface).

Note that an integration board is served as part of the operational environment and is used to provide initial secure initialization of the binary images.

1.4.4 TOE Environment

A group of TOE systems may be interlinked in a network, and individual networks may be joined by bridges and/or routers, or by TOE systems which act as routers and/or gateways. Each TOE system in the network has its own security policy. The TOE does not include any synchronization mechanism for those policies. As a result, a single user may have user accounts with different UIDs and other different attributes on each of those systems. (A method could be used optionally to synchronize these attributes among the systems in the network, but it is not part of the TOE and must not use methods that conflict with the TOE requirements.)

If other systems are connected to the network mentioned above, they need to be configured and managed by the same authority using an appropriate security policy that does not conflict with the security policy of the TOE. All connections between this network and untrusted networks (e. g. the Internet) need to be protected by appropriate measures (e.g. carefully configured firewall systems) that prohibit attacks from the untrusted networks. Those protections are part of the TOE environment.

1.4.5 Security Policy Model

The security policy for the TOE is defined by the security functional requirements in chapter 5. The following is a list of the subjects and objects participating in the policy:

Subjects:

- Processes acting on behalf of a human user.

Named Objects:

- User mode file system objects in the following file systems, which reside on persistent storage devices:
  - **vfat** - standard file system for general data;
  - **jffs2** - flash file system for general data;
- Kernel mode file system objects in the following file systems, which reside in kernel memory (which means the file systems are virtual):
  - **devfs** - temporary file system that allows the kernel to generate character or block device nodes;
  - **procfs** - process file system holding information about processes, general statistical data and tunable kernel parameters;
  - **rootfs** - the virtual root file system used temporarily during system boot;
  - **sysfs** - system-related file system covering general information about resources maintained by the kernel including several tunable parameters for these resources;
  - **tmpfs** - the temporary file system backed by RAM;
  - **disk partition** - logic regions of storage device;
  - **regular file** - a most common file type found on the RTOS V2X system. It governs different files such us text files, images, binary files, shared libraries, etc.
  - **directory** - a special type of file that contains a list a objects (i.e., files, directories and links)

Note that the virtual file systems above implement access decisions based on DAC attributes inferred from that of the underlying process. Additional restrictions may apply for specific objects in them.

- RTOS V2X system call resources:
  - System call number - a unique integer from one to around 256 is assigned to each system call in the operating system.
  - System call parameter – parameters of system call.
- Inter Process Communication (IPC) objects:
  - Named pipes - one-way or duplex pipe for communication between the pipe server and one or more pipe clients.
  - Shared memory - memory shared between two or more processes.
  - Message queues - queue of messages sent between applications. It includes a sequence of work objects that are waiting to be processed.
  - Semaphores - a variable or abstract data type used to control access to a common resource by multiple processes in a concurrent system such as a multitasking operating system.
  - UNIX domain socket special files - A socket is a special file used for inter-process communication, which enables communication between two processes.

Note that named pipes and UNIX sockets exist as special files on disks, and access to them are controlled by the same access control mechanism as to other user mode file system objects.

- Network sockets (irrespectively of their type - such as Internet sockets, *netlink* sockets)

TSF data:

- TSF executable code;
- Subject meta data - all data used for subjects except data which is not interpreted by the TSF and does not implement parts of the TSF (this data is called user data);
- Named object meta data - all data used for the respective objects except data which is not interpreted by the TSF and does not implement parts of the TSF (this data is called user data);
- User accounts, including user identifier, group memberships and user password;
- Audit records;

User data:

- Non-TSF executable code used to drive the behavior of subjects;
- Data not interpreted by TSF and stored or transmitted using named objects;

2. Conformance Claims

This Security Target is CC Part 2 extended and CC Part 3 conformant.

The CC version of [CC] is Version 3.1, Revision 5.

This ST is EAL4 conformance as defined in [CC] Part 3, with the assurance level of EAL4, augmented by ALC_FLR.2.

This Security Target does not claim conformance to any Protection Profile.
Operating System Protection Profile, BSI-CC-PP-0067, version 2.0 and [OSPP-AM] OSPP Extended Package – Advanced Management is taken as a reference.

The TOE type defined in this ST (section 1.3.2) is an embedded operating system, which is in accordance to the TOE type defined in the Protection Profile section 1.2.1.

The security problem definition and the security objectives have been exactly copied from the Protection Profile (section 5 and 6).

All the security requirements defined in the Protection Profile applied to the TOE and have been included in this ST (section 6). All “ST Author Note” and “Application Note” in the Protection Profile have been considered and implemented in the ST, the details are in each SFR and its Application Note in section 6.

The security functionality provided by remote trusted IT system defined in the Protection Profile have been included in this ST (section 6).

3. Security Problem Definition

3.1 Assets

- Persistent storage objects used to store user data and/or TSF data, where this data needs to be protected from any of the following operations:
  - Unauthorized read access
  - Unauthorized modification
  - Unauthorized deletion of the object
  - Unauthorized creation of the new objects
  - Unauthorized management of object attributes
- Transient storage objects, including network data
- TSF functions and associated TSF data
- The resources managed by the TSF that are used to store the above-mentioned objects, including the metadata needed to manage these objects.

3.2 Threats

Threats to be countered by the TOE are characterized by the combination of an asset being subject to a threat, a threat agent and an adverse action.

The definition of threat agents and protected assets that follows is applicable to the OSPP base, unless noted otherwise.

3.2.1 Threat Agents

Threat agents are external entities that potentially may attack the TOE. They satisfy one or more of the following criteria:

- External entities not authorized to access assets may attempt to access them either by masquerading as an authorized entity or by attempting to use TSF services without proper authorization.
- External entities authorized to access certain assets may attempt to access other assets they are not authorized to either by misusing services they are allowed to use or by masquerading as a different external entity.
- Untrusted subjects may attempt to access assets they are not authorized to either by misusing services they are allowed to use or by masquerading as a different subject.

Threat agents are typically characterized by a number of factors, such as expertise, available resources, and motivation, with motivation being linked directly to the value of the assets at stake. The TOE protects against intentional and unintentional breach of TOE security by attackers possessing an enhanced-basic attack potential.

3.2.2 Threats Definition

The table below lists all threats the TOE can counter:

T.ACCESS.TSFDATA	A threat agent might read or modify TSF data without the necessary authorization when the data is stored or transmitted.

T.ACCESS.USERDATA	A threat agent might gain access to user data stored, processed or transmitted by the TOE without being appropriately authorized according to the TOE security policy.
T.ACCESS.TSFFUNC	A threat agent might use or modify functionality of the TSF without the necessary privilege to grant itself or others unauthorized access to TSF data or user data.
T.ACCESS.COMM	A threat agent might access a communication channel that establishes a trust relationship between the TOE and another remote trusted IT system or masquerade as another remote trusted IT system.
T.RESTRICT.NETTRAFFIC	A threat agent might get access to information or transmit information to other recipients via network communication channels without authorization for this communication attempt by the information flow control policy.
T.IA.MASQUERADE	A threat agent might masquerade as an authorized entity including the TOE itself or a part of the TOE in order to gain unauthorized access to user data, TSF data, or TOE resources.
T.IA.USER	A threat agent might gain access to user data, TSF data or TOE resources with the exception of public objects without being identified and authenticated.
T.ROLE.SNOOP	An attacker might obtain the rights granted to a role that was delegated to another user.
T.ROLE.DELEGATE	An attacker might delegate rights granted to a role that he does not possess or that he is not allowed to delegate.

Table 1 Threats countered by the TOE

3.3 Organizational Security Policies

P.ACCOUNTABILITY	The users of the TOE shall be held accountable for their security-relevant actions within the TOE.
P.USER	Authority shall only be given to users who are trusted to perform the actions correctly.
P.APPROVE	Specific rights assigned to users and controlled by the TSF shall only be exercisable if approved by a second user.

Table 2 Organizational Security Policies

3.4 Assumptions

The specific conditions below are assumed to exist in a TOE environment.

3.4.1 Physical aspects

A.PHYSICAL	It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.
A.INTEGRATION_BOARD	The TOE relies on an integration board to provide initial secure initialization of the binary images. Before TOE booting, the integrity of the TOE images should be verified based on image signature via the integration board. After the automatic integrity check is done, TOE images are downloaded to SD6221 SoC based device board, which is AAU5636 in this evaluated TOE

	environment. The integrator must ensure that the TOE images are verified through the integrity check process on the integration board before loading to the SD6221 SoC based device board. This integration board is out of the scope of TOE.

Table 3 Assumption: Physical aspects

3.4.2 Personnel aspects

A.MANAGE	The TOE security functionality is managed by one or more competent individuals. The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the guidance documentation.
A.AUTHUSER	Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.
A.TRAINEDUSER	Users are sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data.

Table 4 Assumption: Personnel aspects

3.4.3 Procedural aspects

A.DETECT	Any modification or corruption of security-enforcing or security-relevant files of the TOE, user or the underlying platform caused either intentionally or accidentally will be detected by an administrative user.
A.PEER.MGT	All remote trusted IT systems trusted by the TSF to provide TSF data or services to the TOE, or to support the TSF in the enforcement of security policy decisions are assumed to be under the same management control and operate under security policy constraints compatible with those of the TOE.
A.PEER.FUNC	All remote trusted IT systems trusted by the TSF to provide TSF data or services to the TOE, or to support the TSF in the enforcement of security policy decisions are assumed to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality.

Table 5 Assumption: Procedural aspects

3.4.4 Connectivity aspects

A.CONNECT	All connections to and from remote trusted IT systems and between physically-separate parts of the TSF not protected by the TSF itself are physically or logically protected within the TOE environment to ensure the integrity and confidentiality of the data transmitted and to ensure the authenticity of the communication end points.

Table 6 Assumption: Connectivity aspects

3.4.5 Security Objectives

3.4.5.1 Security Objectives for the TOE

O.AUDITING	The TSF must be able to record defined security-relevant events (which usually include security-critical actions of users of the TOE). The TSF must protect this information and present it to authorized users if the audit trail is stored on the local system. The information recorded for security-relevant events must contain the time and date the event happened and, if possible, the identification of the user that caused the event, and must be in sufficient detail to help the authorized user detect attempted security violations or potential misconfiguration of the TOE security features that would leave the IT assets open to compromise.
O.CRYPTO.NET	The TSF must allow authorized users to remotely access the TOE using a cryptographically-protected network protocol that ensures integrity and confidentiality of the transported data and is able to authenticate the end points of the communication. Note that the same protocols may also be used in the case where the TSF is physically separated into multiple parts that must communicate securely with each other over untrusted network connections.
O.DISCRETIONARY.ACCESS	The TSF must control access of subjects and/or users to named resources based on identity of the object. The TSF must allow authorized users to specify for each access mode which users/subjects are allowed to access a specific named object in that access mode.
O.NETWORK.FLOW	The TOE shall mediate communication between sets of TOE network interfaces, between a network interface and the TOE itself, and between subjects in the TOE and the TOE itself in accordance with its security policy.
O.SUBJECT.COM	The TOE shall mediate communication between subjects acting with different subject security attributes in accordance with its security policy.
O.I&A	The TOE must ensure that users have been successfully authenticated before allowing any action the TOE has defined to provide to authenticated users only.
O.MANAGE	The TSF must provide all the functions and facilities necessary to support the authorized users that are responsible for the management of TOE security mechanisms and must ensure that only authorized users are able to access such functionality.
O.TRUSTED_CHANNEL	The TSF must be designed and implemented in a manner that allows for establishing a trusted channel between the TOE and a remote trusted IT system that protects the user data and TSF data transferred over this channel from disclosure and undetected modification and prevents masquerading of the remote trusted IT system.
O.ROLE.DELEGATE	The TOE must allow roles assigned to users for performing security-relevant management tasks to be delegated to other users in accordance with the security policy.
O.ROLE.MGMT	The TOE must allow security management actions based on roles to be assigned to different users.

O.ROLE.APPROVE	The TOE must prevent the execution of user actions allowed by a specific right until a second user with a different right approves this action.

Table 7 Security Objectives for the TOE

3.4.5.2 Security Objectives for the Operational Environment

OE.ADMIN	Those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains.
OE.REMOTE	If the TOE relies on remote trusted IT systems to support the enforcement of its policy, those systems provide the functions required by the TOE and are sufficiently protected from any attack that may cause those functions to provide false results.
OE.INFO_PROTECT	Those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner. In particular: ● All network and peripheral cabling must be approved for the transmittal of the most sensitive data held by the system. Such physical links are assumed to be adequately protected against threats to the confidentiality and integrity of the data transmitted. ● DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly. ● Users are authorized to access parts of the data managed by the TOE and are trained to exercise control over their own data.
OE.INSTALL	Those responsible for the TOE must establish and implement procedures to ensure that the hardware, software and firmware components that comprise the system are distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE.
OE.MAINTENANCE	Authorized users of the TOE must ensure that the comprehensive diagnostics facilities provided by the product are invoked at every scheduled preventative maintenance period.
OE.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to enforcement of the security policy are protected from physical attack that might compromise IT security objectives. The protection must be commensurate with the value of the IT assets protected by the TOE.
OE.RECOVER	Those responsible for the TOE must ensure that procedures and/or mechanisms are provided to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.
OE.TRUSTED.IT.SYSTEM	The remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy. These remote trusted IT systems are under the same management domain as the TOE, are managed based on the same rules and policies applicable to the TOE, and are physically and logically

	protected equivalent to the TOE.
OE.INTEGRATION_BOARD	Integrator must integrate an integration board to do secure initialization. The integrity of the TOE should be verified based on image signature. The integration board should be provided by the integrator of the TOE.

Table 8 Security Objectives for the Operational Environment

3.4.5.3 Rationale for Security Objectives

The following tables provide a mapping of security objectives to the environment defined by the threats, policies and assumptions, illustrating that each security objective covers at least one threat, assumption or policy and that each threat, assumption or policy is covered by at least one security objective.

3.4.5.3.1 Security Objectives coverage

Objectives	**SPD coverage**
O.AUDITING	P.ACCOUNTABILITY
O.CRYPTO.NET	T.ACCESS.USERDATA, T.ACCESS.TSFDATA, T.ACCESS.TSFFUNC
O.DISCRETIONARY.ACCESS	T.ACCESS.USERDATA, T.ACCESS.TSFDATA
O.NETWORK.FLOW	T.RESTRICT.NETTRAFFIC
O.SUBJECT.COM	T.ACCESS.USERDATA, T.ACCESS.TSFDATA
O.I&A	T.IA.MASQUERADE, T.IA.USER
O.MANAGE	T.ACCESS.TSFFUNC, P.ACCOUNTABILITY, P.USER
O.TRUSTED_CHANNEL	T.ACCESS.COMM
O.ROLE.DELEGATE	T.ROLE.SNOOP, T.ROLE.DELEGATE
O.ROLE.MGMT	T.ACCESS.TSFFUNC
O.ROLE.APPROVE	P.APPROVE

Table 9 Coverage of security objectives for the TOE

Objectives	SPD coverage
OE.ADMIN	A.AUTHUSER, A.MANAGE, A.TRAINEDUSER
OE.REMOTE	T.ACCESS.COMM, A.CONNECT
OE.INFO_PROTECT	P.USER, A.AUTHUSER, A.TRAINEDUSER, A.PHYSICAL, A.MANAGE
OE.INSTALL	A.MANAGE, A.DETECT
OE.MAINTENANCE	A.DETECT
OE.PHYSICAL	A.PHYSICAL
OE.RECOVER	A.MANAGE, A.DETECT
OE.TRUSTED.IT.SYSTEM	A.CONNECT, A.PEER.MGT, A.PEER.FUNC
OE.INTEGRATION_BOARD	A.INTEGRATION_BOARD

Table 10 Coverage of security objectives for the TOE environment

3.4.5.3.2 Security Objectives sufficiency

Threats	Security Objectives
T.ACCESS.TSFDATA	The threat of accessing TSF data without proper authorization is removed by: ● O.CRYPTO.NET requiring cryptographically-protected communication channels for data including TSF data controlled by the TOE in transit between trusted IT systems, ● O.DISCRETIONARY.ACCESS requiring that data, including TSF data stored with the TOE, have discretionary access control protection, ● O.SUBJECT.COM requiring the TSF to mediate communication between subjects.
T.ACCESS.USERDATA	The threat of accessing user data without proper authorization is removed by: ● O.CRYPTO.NET requiring cryptographically-protected communication channels for data including user data controlled by the TOE in transit between trusted IT systems, ● O.DISCRETIONARY.ACCESS requiring that data including user data stored with the TOE, have discretionary access control protection, ● O.SUBJECT.COM requiring the TSF to mediate communication between subjects.

T.ACCESS.TSFFUNC	The threat of accessing TSF functions without proper authorization is removed by: ● O.MANAGE requiring that only authorized users utilize management TSF functions. ● O.CRYPTO.NET requiring cryptographically-protected communication channels to limit which TSF functions are accessible to external entities. ● O.ROLE.MGMT requiring the TOE to allow security management actions based on roles to be assigned to different users.
T.ACCESS.COMM	The threat of accessing a communication channel that establishes a trust relationship between the TOE and another remote trusted IT system is removed by: ● O.TRUSTED_CHANNEL requiring that the TOE implements a trusted channel between itself and a remote trusted IT system protecting the user data and TSF data transferred over this channel from disclosure and undetected modification and prevents masquerading of the remote trusted IT system, ● OE.REMOTE requiring that those systems providing the functions required by the TOE are sufficiently protected from any attack that may cause those functions to provide false results.
T.RESTRICT.NETTRAFFIC	The threat of accessing information or transmitting information to other recipients via network communication channels without authorization for this communication attempt is removed by: ● O.NETWORK.FLOW requiring the TOE to mediate the communication between itself and remote entities in accordance with its security policy.
T.IA.MASQUERADE	The threat of masquerading as an authorized entity in order to gain unauthorized access to user data, TSF data or TOE resources is removed by: ● O.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE is defined to provide to authenticated users only.
T.IA.USER	The threat of accessing user data, TSF data or TOE resources without being identified and authenticated is removed by: ● O.I&A requiring that each entity interacting with the TOE is properly identified and authenticated before allowing any action the TOE has defined to provide to authenticated users only.
T.ROLE.SNOOP	The threat of an attacker obtaining the rights granted to a role that was delegated to another user is removed by: ● O.ROLE.DELEGATE requiring the TOE to allow delegation of roles to other users in accordance with the security policy.
T.ROLE.DELEGATE	The threat of an attacker delegating rights granted to a role that he does not possess or that he is not allowed to delegate is removed by: ● O.ROLE.DELEGATE requiring the TOE to allow roles assigned to users for performing security-relevant management tasks to be delegated.

Table 11 TOE threats sufficiency

Security Policies	Security Objectives
P.ACCOUNTABILITY	The policy to hold users accountable for their security-relevant actions within the TOE is implemented by: ● O.AUDITING providing the TOE with audit functionality, ● O.MANAGE allowing the management of this function.
P.USER	The policy to match the trust given to a user and the actions the user is given authority to perform is implemented by: ● O.MANAGE allowing appropriately-authorized users to manage the TSF, ● OE.INFO_PROTECT, which requires that users are trusted to use the protection mechanisms of the TOE to protect their data.
P.APPROVE	The policy that specific rights assigned to users shall only be exercisable when approved by a second user is implemented by: ● O.ROLE.APPROVE requiring the TOE to prevent the execution of user actions allowed by a specific right until a second user with a different right approves this action.

Table 12 Security policies sufficiency

Assumptions	Security Objectives
A.PHYSICAL	The assumption on the IT environment to provide the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE is covered by: ● OE.INFO_PROTECT requiring the approval of network and peripheral cabling, ● OE.PHYSICAL requiring physical protection.
A.MANAGE	The assumptions on the TOE security functionality being managed by one or more trustworthy individuals is covered by: ● OE.ADMIN requiring trustworthy personnel managing the TOE, ● OE.INFO_PROTECT requiring personnel to ensure that information is protected in an appropriate manner, ● OE.INSTALL requiring personnel to ensure that components that comprise the system are distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE, ● OE.RECOVER requiring personnel to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.

A.AUTHUSER	The assumption on authorized users to possess the necessary authorization to access at least some of the information managed by the TOE and to act in a cooperating manner in a benign environment is covered by: ● OE.ADMIN ensuring that those responsible for the TOE are competent and trustworthy individuals, capable of managing the TOE and the security of the information it contains, ● OE.INFO_PROTECT requiring that DAC protections on security-relevant files (such as audit trails and authentication databases) shall always be set up correctly and that users are authorized to access parts of the data maintained by the TOE.
A.TRAINEDUSER	The assumptions on users to be sufficiently trained and trusted to accomplish some task or group of tasks within a secure IT environment by exercising complete control over their user data is covered by: ● OE.ADMIN requiring competent personnel managing the TOE, ● OE.INFO_PROTECT requiring that those responsible for the TOE must establish and implement procedures to ensure that information is protected in an appropriate manner and that users are trained to exercise control over their own data.
A.DETECT	The assumption that modification or corruption of security-enforcing or security-relevant files will be detected by an administrative user is covered by: ● OE.INSTALL requiring an administrative user to ensure that the TOE is distributed, installed and configured in a secure manner supporting the security mechanisms provided by the TOE, ● OE.MAINTENANCE requiring an administrative user to ensure that the diagnostics facilities are invoked at every scheduled preventative maintenance period, verifying the correct operation of the TOE, ● OE.RECOVER requiring an administrative user to ensure that procedures and/or mechanisms are provided to assure that after system failure or other discontinuity, recovery without a protection (security) compromise is achieved.
A.PEER.MGT	The assumption on all remote trusted IT systems to be under the same management control and operate under security policy constraints compatible with those of the TOE is covered by: ● OE.TRUSTED.IT.SYSTEM requiring that these remote trusted IT systems are under the same management domain as the TOE, and are managed based on the same rules and policies applicable to the TOE.
A.PEER.FUNC	The assumption on all remote trusted IT systems to correctly implement the functionality used by the TSF consistent with the assumptions defined for this functionality is covered by: ● OE.TRUSTED.IT.SYSTEM requiring that the remote trusted IT systems implement the protocols and mechanisms required by the TSF to support the enforcement of the security policy.

A.CONNECT	The assumption on all connections to and from remote trusted IT systems and between physically separate parts of the TSF not protected by the TSF itself are physically or logically protected is covered by: ● OE.REMOTE requiring that remote trusted IT systems provide the functions required by the TOE and are sufficiently protected from any attack that may cause those functions to provide false results, ● OE.TRUSTED.IT.SYSTEM demanding the physical and logical protection equivalent to the TOE.
A.INTEGRATION_BOARD	The assumption on the existence of an integration board used to conduct initial secure initialization by verifying signatures of the TOE binary image is covered by: ● OE.INTEGRATION_BOARD requiring the integrator to integrate an integration board to do secure initialization. The integrity of the TOE should be verified based on image signature.

Table 13 Assumptions sufficiency

4. Extended Components Definition

These extended components are defined in [OSPP] Extended components definition section.

4.1 FDP_RIP.3 Full residual information protection of resources

FDP_RIP.3 is analog to FDP_RIP.2 except that it applies to the content of resources that are allocated to subjects or users.

4.1.1 Component leveling

FDP_RIP.3 is not hierarchical to any other component within the FDP_RIP family.

4.1.2 Management

See management description specified for FDP_RIP.2 in [CC2].

4.1.3 Audit

See audit requirement specified for FDP_RIP.2 in [CC2].

4.1.4 FDP_RIP.3 Full residual information protection of resources

Hierarchical to: No other component
Dependencies: No dependencies
FDP_RIP.3.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: allocation of the resource to, de-

allocation of the resource from] all subjects or users.

4.1.5 Rationale

FDP_RIP.3 addresses the problem of resources implemented in main memory that may be allocated to and de-allocated from subjects or users. Unless those resources lose their content automatically as part of the de-allocation and re-allocation process, they must be subject to a process that prepares them for re-use by rendering the previous content unavailable to the subject or user to which it is next allocated. An example is main memory that has been allocated to a subject; this memory must be cleared before it can be re-allocated to a subject with different security attributes (for example a subject operating on behalf of a different user). This preparation prevents the passing of security-critical information via this resource, since such unregulated passing would potentially allow the subject or user to which the memory is next allocated to use this information to violate the security policy. Typical examples of such critical information that may be passed via resources not prepared for re-use are passwords or cryptographic keys.

4.2 FIA_USB.2 Enhanced user-subject binding

FIA_USB.2 is analog to FIA_USB.1 except that it adds the possibility to specify rules whereby subject security attributes are also derived from TSF data other than user security attributes.

4.2.1 Component leveling

FIA_USB.2 is hierarchical to FIA_USB.1.

4.2.2 Management

See management description specified for FIA_USB.1 in [CC2].

4.2.3 Audit

See audit requirement specified for FIA_USB.1 in [CC2].

4.2.4 FIA_USB.2 Enhanced user-subject binding

Hierarchical to: FIA_USB.1 User-subject binding
Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.2.1	The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: list of user security attributes].
FIA_USB.2.2	The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: rules for the initial association of attributes].
FIA_USB.2.3	The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: rules for the changing of attributes].
FIA_USB.2.4	**The TSF shall enforce the following rules for the assignment of**

	subject security attributes not derived from user security attributes when a subject is created: [assignment: rules for the initial association of the subject security attributes not derived from user security attributes].

4.2.5 Rationale

An operating system may derive subject security attributes from other TSF data that are not directly user security attributes. An example is the point-of-entry the user has used to establish the connection. An access control policy may also use this subject security attribute within its access control policy, allowing access to critical objects only when the user has connected through specific ports-of-entry.

5. Security Requirements

The TOE is an embedded operating system and OSPP is followed as a reference. Following typographical conventions are used for marking operations in this ST:

- Assignments and selections are marked in **bold** face font.
- Iterations are marked by appending a suffix to the SFR identification.
- Refinements are marked in ***bold and italic*** face font.
- ~~Strikethrough~~ indicates text removed as a refinement.

5.1 Security Functional Requirements

5.1.1 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 shutdown of the audit functions;
b) All auditable events for the **basic** level of audit; and
c) **all modifications to the set of events being audited;**
all user authentication attempts;
all denied accesses to objects for which the access control policy defined in the ST applies;
explicit modifications of access rights to objects covered by the access control policies;
no other auditable events

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 outcome of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the ST;
   i. **User identity (if applicable); and**
   ii. **none**

Application Note: The subject identity may be identical to the user identity in the case where the subject identity is established by the user-subject binding process. In this case, only one identity needs to be included in the audit record. The purpose here is the ability to trace an event to the user that caused the event. This may not be possible if the subject identity does not allow to identify the user the subject was bound to when the event happened. In order to support FAU_GEN.2, the user identity has, therefore, been added as the information to be recorded.

Application Note: The outcome to be recorded with the audited event can either be binary (success or failure) or the value resulting from the event, depending on the implementation of the TOE. For example, access control decision shall store the information about the result of the access control decision with the audit trail. A TOE may implement more decision results than just access allowed or denied, where all of these results shall be recorded as outcome of the access control check event.

5.1.2 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 caused the event.

5.1.3 FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide **the root user** with the capability to read **all audit information defined in FAU_GEN.1.1** 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.

Application Note: The root user, also known as the super user or administrator, is a special user account in RTOS V2X used for system administration.

Application Note: The TOE may store its audit records locally, or it may pass its audit records on to a remote trusted IT system for storage and further processing. The trusted IT system also applies to FAU_SAR.1.

5.1.4 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.

5.1.5 FAU_SEL.1 Selective audit

FAU_SEL.1.1 The TSF shall be able to select the set of events to be audited from the set of all auditable events based on the following attributes:

a) Subject or user identity;
Named object identity;
b) Type of audit event;
Outcome (success or failure) of the audit event;
Access type to file system objects (read, write, execute, change attributes);
c) System call number.

Application Note: The TOE provides an application that allows specification of the audit rules which injects the rules into the kernel for enforcement. The RTOS V2X kernel auditing mechanism obtains all audit events and decides based on this rule set whether an event is forwarded to the audit daemon for storage.

5.1.6 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 audit records in the audit trail.

Application Note: The TOE may store its audit records locally, or it may pass its audit records on to a remote trusted IT system for storage and further processing. Even in this case, the TOE will usually need some kind of local audit trail as a (probably volatile) cache to buffer some audit records or to bridge the time when the remote audit server might not be available. Such a local audit trail must be protected as described in this SFR.

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

FAU_STG.3.1 The TSF shall **notify an authorized administrator** if the audit trail exceeds **a root-user selectable, pre-defined size limit of the audit trail** ~~***or if any of the following [assignment: no other condition] is detected that may result in a loss of audit records.***~~

Application Note The term "authorized administrator" refers to the user that is notified by the *auditd* daemon. This daemon can be configured to notify different users in different ways. The administrator of the system must ensure that the *auditd* is configured to send the notification to the intended recipient.

Application Note The alarm generated by the TOE can be configured to be a syslog message or the execution of an administrator-specified application. This message or action of executing the application is generated when the audit trail capacity exceeds the limit defined in the *auditd.conf* file.

Application Note The information of the threshold limit is done in the configuration file of the *auditd* daemon. This file is only writable to the root user.

5.1.8 FAU_STG.4 Prevention of audit data loss

FAU_STG.4.1 The TSF shall **overwrite the oldest stored audit records** and **issue warnings in case of audit storage failure** if the audit trail is full.

Application Note: When the audit trail is full (i.e. the disk is full), the TOE should overwrite the oldest stored audit records. If audit storage failed, the TOE should send warnings to syslog.

5.1.9 FCS_CKM.1(SYM) Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **supported by SSH/TLS** and specified cryptographic key sizes:

- **a) 128 bits,**
- **b) 168 bits,**
- **c) 256 bits,**
- **d) 192 bits,**
- **e) 384 bits,**
- **f) 512 bits,**

supported by SSH/TLS that meet the following:

- **a) SSH: generation and exchange of session keys using the Diffie-Hellman key negotiation protocol as defined in RFC4253;**
- **b) TLS: generation and exchange of session keys as defined in the TLSV1.2 standards with the cipher suites defined in FCS_COP.1(NET);**

5.1.10 FCS_CKM.1(RSA) Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate ***RSA*** cryptographic keys in accordance with a specified cryptographic key generation algorithm **defined in U.S. NIST FIPS PUB ~~186-3~~ *186-4* appendix B.3** and specified cryptographic key sizes:

- **a) 2048 bits,**
- **b) 3072 bits**

that meet the following:

a) **U.S. NIST FIPS PUB ~~186-3~~ *186-4* [1],**

b) **no other standards**

Application Note: The TOE supports the generation of RSA keys for the OpenSSH host key as well as the OpenSSH user keys using the *ssh-keygen(1)* application. Also, an internal non-physical true random generator is used to support the key generation.

5.1.11 FCS_CKM.1(DSA) Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate ***DSA*** cryptographic keys in accordance with a specified cryptographic key generation algorithm **defined in U.S. NIST FIPS PUB ~~186-3~~ *186-4* appendix B.1** and specified cryptographic key sizes:

a) **L=2048, N=256 bits;]**

that meet the following:

a) **U.S. NIST FIPS PUB ~~186-3~~ *186-4*[2].**

b) **no other standards**

Application Note: The TOE supports the generation of DSA keys for the TLS protocols using the *openssl(1)* application.

5.1.12 FCS_CKM.2(NET) Cryptographic key distribution

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with ***~~a~~ the following*** specified cryptographic key distribution method ***~~[assignment: cryptographic key distribution method]~~*** that meets the following:

a) Diffie-Hellman key agreement method defined for the SSH protocol by RFC4253 *together with RFC8268*: *diffie-hellman-group14-sha256*;

b) Public RSAhost key exchange defined for the SSH protocol by RFC4253;

c) RSA encrypted exchange of pre-master secrets defined for the TLS protocol by RFC5246;

d) Diffie-Hellman key agreement method defined for the SSH protocol by RFC4253 together with RFC4419 and RFC8268: diffie-hellman-group-exchange-sha256;

e) EC Diffie-Hellman key agreement method defined for the SSH protocol by RFC4253 together with [RFC5656]: with ecdh-sha2-nistp521].

5.1.13 FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **of zeroization** that meets the following: **vendor-specific zeroization**.

Application Note: The "vendor-specific zeroization" covers to the following concepts:

- Memory objects: Overwriting the memory with zeros at the time the memory is released.
- Asymmetric key components stored in files: The object reuse functionality for objects defined with FDP_RIP.2 also covers this SFR.

5.1.14 FCS_COP.1(NET) Cryptographic operation

[1] OSPP requires FIPS PUB 186-3, which is superseded by the new version FIPS PUB 186-4.

[2] OSPP requires FIPS PUB 186-3, which is superseded by the new version FIPS PUB 186-4.

FCS_COP.1.1 The TSF shall perform **encryption, decryption, integrity verification, peer authentication** in accordance with with ***~~a specified~~ the following*** cryptographic algorithms, cryptographic key sizes ***~~[assignment: cryptographic key sizes] that meet the following~~ and applicable standards***:

***a*) SSH allowing the use of AES in CBC mode with 128 bits and 256 bits key size, and ~~HMAC-SHA1~~ *HMAC-SHA2-256* defined by RFC 4253 *with additional definition in RFC6668*;**

b) other cryptographic network algorithms, keys sizes with their standards:

- **SSH communication channel encryption using the following ciphers as defined in [RFC4253]:**
  1. **AES in CBC mode (aes192-cbc);**
  2. **AES in CTR mode (aes128-ctr, aes192-ctr, aes256-ctr) with additional definition in [RFC4344];**
  3. **AES in GCM mode (AEAD_AES_128_GCM, AEAD_AES_256_GCM) with additional definition in [RFC 5647];**
  4. **HMAC with SHA-2 (hmac-sha2-256, hmac-sha2-512, hmac-sha2-256-etm@openssh.com, hmac-sha2-512-etm@openssh.com) with additional definition in [RFC6668]**
- **SSH authentication of user as defined in [RFC4252] together with [RFC8332]:**
  1. **RSA signature verification RSASSA-PKCS1-v1.5 using SHA-2 (rsa-sha2-256).**

5.1.15 FDP_ACC.1(PSO) Subset access control

FDP_ACC.1.1 The TSF shall enforce the **Persistent Storage Object Access Control Policy** on:

a) **Subjects: all subjects defined with the Security Policy Model**

b) Objects:

i. **Persistent Storage Objects of the following type:** [**all file system objects defined with the Security Policy Model**];

ii. **no other storage objects;**

c) **Operations**: **read, write, execute (regular files), search (permission to change to the directory, or to search a file/directory under the directory).**

Application Note: A persistent storage object establishes a data storage or data exchange link between two or more subjects. Examples of persistent storage objects are: files, directories.

Application Note: The creation of an object under a directory, write permission is required to the directory; The destruction of an object under a directory, write permission is required to both the directory and the object (without considering the sticky bit); If the sticky bit is considered, only the owner of the object can destroy the object. The operations on TSF data associated and stored with the object are using discretionary access control, only the owner of the file or administrator can modify security attributes.

Application Note: The information of "Security Policy Model" please refers to chapter 1.4.5 Security Policy Model.

5.1.16 FDP_ACC.1(TSO) Subset access control

FDP_ACC.1.1 The TSF shall enforce the **Transient Storage Object Access Control Policy** on:

a) **all subjects defined with the Security Policy Model;**
b) **Objects:**
   i. **Transient Storage Objects of the following type: all IPC objects defined with the Security Policy Model;**
   ii. **no other storage objects;**
c) **Operations: read (includes receive), write (includes send)**.

Application Note: A transient storage object establishes a data exchange link between two or more subjects or users. Examples of transient storage objects are: shared memory, semaphores, message queues, named pipes.

Application Note: Security Policy Model please refers to chapter 1.4.5 Security Policy Model.

5.1.17 FDP_ACF.1(PSO) Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the **Persistent Storage Object Access Control Policy** to objects based on the following:

a) Subject security attributes: file system UID, file system GID, supplementary GIDs;
b) Object security attributes: owning UID, owning GID;
c) Access control security attributes maintained for each file system object governing access to that object:
   i. Permission bits for the owning UID,
   ii. Permission bits for the owning GID,
   iii. Permission bits for all other users (“world”),
   iv. The following permission bits: read, write, execute (for files), search (for directories),
   v. The following access rights applicable to the file system object: SAVETXT (sticky bit for directories),
   vi. The following capabilities: CAP_DAC_READ_SEARCH, CAP_DAC_OVERRIDE, CAP_FOWNER;
d) **Access control security attributes maintained for each disk partition that holds a file system: read-only, no-execute.**

Application Note: All object security attributes associated with each of the different file system objects of Security Policy Model. For file system objects, please refer to section 1.4.5 Security Policy Model.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

A subject must have search permission for every element of the pathname and the requested access for the object. A subject has a specific type access to an object if one of the following rules hold:

a) The subject's filesystem UID is identical with the owning UID of the object and the requested type of access is within the permission bits defined for the owning UID (permission bits); or
b) The subject's filesystem GID or one of the subject's supplementary GIDs is identical with the owning GID and the requested type of access is within the permission bits defined for the owning GID (permission bits); or
c) **The requested type of access is within the permission bits defined for "all other users" (permission bits).**

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

a) read and directory search operations are allowed for the subject with the capability of CAP_DAC_READ_SEARCH;

b) **write and execute operations are allowed for the subject with the capability of CAP_DAC_OVERRIDE - the execute permission is granted if the file system object is marked with at least one executable bit in its permission settings.**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to named objects based on the following additional rules:

a) Any file system object in a file system that is mounted as read-only cannot be modified, created or removed;

b) A regular file, a directory and a symbolic link in a file system that is mounted as read-only cannot be written to;

c) A regular file in a file system that is mounted with the no-execute flag cannot be executed;

d) **Any file system object stored in a directory marked with the SAVETXT(sticky bit for directories) cannot be modified or removed by subjects whose file system UID is not equal to the owning UID of the file system object unless the subject performing the operation possesses the CAP_FOWNER capability.**

Application note: Although a persistent file system, *vfat* is not POSIX-compliant and hence does not support the access control policy for each single file/directory. However, the security attributes supporting the access control policy can be specified uniformly for all files/directories *as a whole* at mount time, such as uid=uuu,gid=ggg,fmask=0000,dmask=0000. As a result, command /bin/chown always fails for vfat objects, and command /bin/chmod seems to work well on vfat objects, but it does not change anything.

5.1.18 FDP_ACF.1(TSO) Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the **Transient Storage Object Access Control Policy** to objects based on the following:

a) Subject security attributes: effective UID, file system UID, effective GID, file system GID, supplementary GIDs;

b) Object security attributes: owning UID, owning GID;

c) Access control security attributes maintained for each IPC object whose name is managed with a file governing access to that object: see FDP_ACF.1(PSO);

d) Access control security attributes maintained for any other IPC object governing access to that object:

i. Permission bits for the owning UID,

ii. Permission bits for the owning GID,

iii. Permission bits for "world",

iv. The following permission bits: read, write, execute,

v. **The following capabilities: CAP_IPC_FOWNER.**

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

a) IPC object whose name is managed with a file: see FDP_ACF.1(PSO);

b) Any other IPC object: A subject has a specific type access to an object if one of the following rules hold:

1. The subject's effective UID is identical with the owning UID of the

object and the requested type of access is within the permission bits defined for the owning UID; or

2. **The subject's effective GID or one of the subject's supplementary GIDs is identical with the owning GID and the requested type of access is within the permission bits defined for the owning GID; or**
3. **The requested type of access is within the permission bits defined for "world".**

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

a) IPC object whose name is managed with a file: see FDP_ACF.1(PSO);

b) Any other IPC object:

1. **read, write, send and receive operations are allowed for the subject with the capability of CAP_IPC_OWNER.**

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to named objects based on the following rules:

a) IPC object whose name is managed with a file: see FDP_ACF.1(PSO);

b) **Any other IPC object: none.**

Application Note: All object security attributes associated with the following IPC objects: Named pipes; Shared memory; Message queues; Semaphores.

5.1.19 FDP_IFC.2(NI) Complete information flow control

FDP_IFC.2.1 The TSF shall enforce the **Network Information Flow Control Policy** on:

a) Subjects:

i. unauthenticated external IT entities that send and receive information mediated by the TOE;

ii. RTOS V2X processes that send and receive information mediated by the TOE;

b) Information:

i. Network data routed through the TOE;

ii. Network data received by the TOE from an external IT entity;

iii. **Network data provided to the TOE by a subject executing on the TOE intended to be sent to an external IT entity via a network interface controlled by the TOE.**

and all operations that cause that information to flow to and from subjects covered by the SFP.

FDP_IFC.2.2 The TSF shall ensure that all operations that cause any information in the TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

5.1.20 FDP_IFF.1(NI) Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the **Network Information Flow Control Policy** based on the following types of subject and information security attributes:

a) ~~Object~~ *Information* security attribute: the logical or physical network interface through which the network data entered the TOE;

b) TCP/IP information security attributes:

i. Source and destination IP address,

ii. Source and destination TCP port number,

iii. Source and destination UDP port number,

iv. Network protocol of TCP, UDP, ICMP, no other protocols

v. TCP header flags of [selection: SYN, ACK, FIN, RST, URG, PSH,

vi. no other attributes;

Application Note: Logical network interfaces include the interface provided by the TOE to local subjects acting on behalf of local users. Such interfaces may include network sockets introduced by the Berkeley Software Distribution (BSD) or any other mechanism that allows subject to initiate an IP-based connection.

Application Note: The minimum requirement of the network flow control specified in FDP_IFF.1.3(NI) defines the purpose of the Network Information Flow Control Policy, namely to identify network data using the security attributes specified here and to at least discard the identified network data or allow it to pass the TOE unaltered.

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **If the packet filter matches the analyzed packet and the rule described in FDP_IFF.1.3 accepts the packet, the packet is forwarded according to the network protocol stack's behavior.**

FDP_IFF.1.3 The TSF shall enforce **the following rules:**

Identification of network data using one or more of the following concepts:

a) Information security attribute matching;

b) Matching based on the state of a TCP connection, no other matching concepts;

Performing one or more of the following actions with identified network data:

a) Discard the network data without any further processing, with sending a notification to the sender;

b) Allow the network data to be processed unaltered by the TOE according to the routing information maintained by the TOE;

c) **no other actions.**

FDP_IFF.1.4 The TSF shall explicitly authorize an information flow based on the following rules: **If the network data is not matched by the rule set and the default rule of the packet filter is ACCEPT then the data is forwarded unaltered based on the normal operation of the host system's networking stack.**

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: **If the network data is not matched by the rule set, one of the following default rules applies:**

a) DROP: the data is discarded.

Application Note: The default rule is configurable, could either be accept, drop, forward or reject. The default rules can be selected at any given time.

Application Note: This requirements covers IPv4 traffic.

5.1.21 FDP_ITC.2 Import of user data with security attributes

FDP_ITC.2.1 The TSF shall enforce the **Persistent Storage Access Control Policy**, **Transient Storage Access Control Policy**, **Network Information Flow Control**, **no other access control SFP(s) and/or information flow control SFP(s)** when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: **No additional importation control rules.**

Application Note: Based on the wording of FDP_ITC.2.1, the TOE complies with this SFR even when it does not allow import of objects covered by the persistent or transient storage object control policy.
However, the network information flow control policy must always be covered by the TOE, as it applies to the networking capability of the TOE to control traffic originating from outside the TOE. In this case, the interpretation of security attributes is defined by the respective protocol family.

5.1.22 FDP_RIP.2 Full residual information protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** all objects.

5.1.23 FDP_RIP.3 Full residual information protection of resources

FDP_RIP.3.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **allocation of the resource to** all subjects or users.

5.1.24 FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when **an administrator-configurable number of** unsuccessful authentication attempts ***for the authentication method of password-based authentication*** occur related to **consecutive unsuccessful authentication attempts**.

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

a) For all accounts, "disable" the account for a time period configured by the administrator.

b) **For all disabled accounts, any response to an authentication attempt given to the user shall not be based on the result of that authentication attempt.**

5.1.25 FIA_ATD.1(HU) User attribute definition

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

a) User identifier;
b) Group memberships;
c) User password;
d) Software token verification data;
e) Security roles;
f) **no other user security attributes**

Application Note: The TOE is able to maintain the following types of software tokens and their verification data:
SSH user keys: The TOE as server part is able to store the public part of the SSH user key for the user account the user wants to access. When the TOE acts as an SSH client, the TOE is able to store the private part of the SSH user key for the

requesting user.

Application Note: See the application note for FIA_UAU.5 for a list of token-based authentication mechanisms and their associated tokens.

5.1.26 FIA_ATD.1(TU) User attribute definition

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

a) **the logical or physical network interface through which the network data entered the TOE;**
b) **identity of the logical or physical external interface through which the user connected to the TOE;**
c) **no other user security attributes**

Application Note: Bullet a) of this SFR relates to FDP_IFC.2(NI) and FDP_IFF.1(NI). In the Common Criteria scheme, external entities are always considered to be users. Therefore, every network data entity must be specified as user in this ST.

5.1.27 FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet **the following quality metric: the probability that a secret can be obtained by an attacker during the lifetime of the secret is less than 2^(-20).**

Application Note: The TOE password change is implemented using the PAM library. The PAM module *pam_cracklib.so* allows the specification of the quality of new passwords. The evaluated configuration requires a configuration of the PAM-based password change mechanism that meets the above mentioned criteria.

Application Note: For key-based authentication methods, the evaluation of the RSA, DSA keys used for the SSH protocol will show the maximum lifetime of a key depending on its size.

5.1.28 FIA_UAU.1 Timing of authentication

FIA_UAU.1.1 The TSF shall allow

a) the information flow covered by the Network Information Flow Control Policy;
b) Establishing a cryptographically secured network connection;
c) Local console log-in: banner information;
d) SSH log-in: obtaining the list of allowed authentication methods;

on behalf of the user to be performed before the user is authenticated.

Application note The banner for local login and remote login is in file /etc/motd. The banner for remote login can be set in file /etc/ssh/sshd_config.

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

5.1.29 FIA_UAU.5 Multiple authentication mechanisms

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

a) Authentication based on username and password;
b) Authentication based on software token verification data;

c) no other authentication mechanisms

to support user authentication.

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

a) Authentication based on username and password is performed for TOE-originated requests and credentials stored by the TSF;

b) Authentication based on software token verification data is performed for TOE-originated requests;

c) For SSH, both the password-based and key-based authentication methods can be enabled at the same time. In this case, the key-based authentication method is tried before the password-based authentication. If the key-based authentication succeeds, the user is authenticated. If the key-based authentication fails, the password-based authentication is applied. If the password-based authentication fails, the user login request is denied. Additionally, Users with expired passwords are required to create a new password after correctly entering the expired password.

Application Note: For the term "software token verification data", see the application note for FIA_ATD.1(HU).

Application Note: For username and password authentication scenario, the login application uses PAM api to do authentication; for software token authentication scenario, sshd verifies the auth data provided by ssh client.

5.1.30 FIA_UAU.7 Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only **obscured feedback** to the user while the authentication is in progress.

5.1.31 FIA_UID.1 Timing of identification

FIA_UID.1.1 The TSF shall allow

a) the information flow covered by the Network Information Flow Control Policy;

b) Establishing a cryptographically secured network connection;

c) Console log-in: banner information;

d) **SSH log-in: obtaining the list of allowed authentication methods;**

on behalf of the user to be performed before the user is identified.

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

5.1.32 FIA_USB.2 Enhanced user-subject binding

FIA_USB.2.1 The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

a) The user identity that is associated with auditable events;

b) The user security attributes that are used to enforce the Persistent Storage Object Access Control Policy;

c) The user security attributes that are used to enforce the Transient Storage Object Access Control Policy;

d) The software token that can be used for subsequent identification and authentication with the TSF or other remote IT systems;

e) Active roles;

f) Active groups;

g) no other security attributes

FIA_USB.2.2 The TSF shall enforce the following rules on the initial association of security attributes with subjects acting on the behalf of users:

a) Upon successful identification and authentication, the login UID, the real UID, the filesystem UID and the effective UID shall be those specified in the user entry for the user that has authenticated successfully;

b) Upon successful identification and authentication, the real GID, the filesystem GID and the effective GID shall be those specified via the primary group membership attribute in the user entry;

c) Upon successful identification and authentication, the supplementary GIDs shall be those specified via the supplementary group membership assignment for the user entry;

Application Note: The various subject UIDs are all derived from the same numeric UID per user entry stored in the */etc/passwd* file. The subject's supplementary GIDs are derived from the username to group name mappings in the */etc/group* file. As the TOE only maintains numeric IDs for subjects, the username and the group names need to be converted before instantiating the subject. The username to UID mapping is provided in */etc/passwd* file and the group name to GID mapping is provided in */etc/group* file.

FIA_USB.2.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

a) The effective and filesystem UID of a subject can be changed by the use of an executable with the SETUID bit set. In this case the program is executed with the effective and filesystem UID of the owning UID of the file storing the program. These newly set effective and filesystem UIDs are used for the DAC permission validation. The real and login UID remain unchanged.

b) The effective and filesystem GID of a subject can be changed by the use of an executable with the SETGID bit set. In this case the program is executed with the effective and filesystem GID of the owning GID of the file storing the program. These newly set effective and filesystem GIDs are used for the DAC permission validation. The real GID remains unchanged.

c) The real, effective and filesystem UID of a subject can be changed by the use of the *set*uid* system call family for the calling application. These system calls are restricted to processes possessing the *CAP_SETUID* capability.

d) The real, effective and filesystem GID of a subject can be changed by the use of the *set*gid* system call family for the calling application. These system calls are restricted to processes possessing the *CAP_SETGID* capability.

e) The set of supplementary GIDs of a subject can be changed by the use of the *setgroups* system call for the calling application. These system calls are restricted to processes possessing the *CAP_SETGID* capability.

f) The set of effective and inheritable capabilities of a subject can be changed by the use of an executable with activated file capabilities. In this case the program obtains the following capabilities when invoking the file with *execve*:

1. the process' effective capability set gains the capabilities defined by the permitted file capabilities set;

2. **the process' inheritable capability set is ANDed with the inheritable**

file capability set to form the new process' inheritable capability set which defines the capability set that will be retained after an *execve* system call.

FIA_USB.2.4 The TSF shall enforce the following rules for the assignment of subject security attributes not derived from user security attributes when a subject is created: **no rules**.

Application Note: The application "su" allows the calling user to change the filesystem and effective UID either to root or to other users provided the authentication to "su" was successful. The application uses the SETUID bit with the owning UID of root as well as the set*uid system calls to change to other UIDs before spawning a new shell or the given command. As the application rests on the above mentioned mechanisms, it is not listed as a separate mechanism to modify the calling user's UIDs.

Application Note: The login UID is set by the PAM modules by inserting the intended UID into the */proc/<PID>/loginuid* file. This file can be written to only by subjects executing with the effective UID of zero (root) and only for the calling process' own *loginuid* file. However, there is no application except the PAM modules which access that proc file which implies that the login UID remains unchanged after login when operating the TOE. Authorized administrators are not intended to access that proc file.

5.1.33 FMT_MSA.1(PSO) Management of object security attributes

FMT_MSA.1.1 The TSF shall enforce the **Persistent Storage Object Access Control Policy** to restrict the ability to **modify**, **change_default** the security attributes **of the objects covered by the SFP** to **the owner of the object and users with processes granted the *CAP_CHOWN, CAP_FOWNER, CAP_FSETID* capabilities**.

5.1.34 FMT_MSA.1(TSO) Management of object security attributes

FMT_MSA.1.1 The TSF shall enforce the **Transient Storage Object Access Control Policy** to restrict the ability to **modify**, **change_default** the security attributes **of the objects covered by the SFP** to **the owner of the object and users with processes granted the *CAP_SYS_ADMIN* capability**.

5.1.35 FMT_MSA.3(PSO) Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **Persistent Storage Object Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow

a) administrators for a global setting applied during logon;

b) **users for a setting applicable to his processes;**

to specify alternative initial values to override the default values when an object or information is created.

Application Note: The global default value for permission bits is specified with the *umask* value which specifies the permission bits for newly created objects. This value has an initial setting of *022* or the value specified in */etc/login.defs*. Only the root user can manage that initial value as this file is writable to root only. Users can change their umask value at any time using the *umask(2)* system call.

5.1.36 FMT_MSA.3(TSO) Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **Transient Storage Object Access Control Policy** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the

a) administrator for a global setting applied during logon;

b) users for a setting applicable to his processes

to specify alternative initial values to override the default values when an object or information is created.

Application Note: The global default value for permission bits is specified with the *umask* value which specifies the permission bits for newly created objects. This value has an initial setting of *022* or the value specified in */etc/login.defs*. Only the root user can manage that initial value as this file is writable to root only. Users can change their *umask* value at any time using the *umask(2)* system call.

NOTE: *umask* does not impact on System V IPC objects, and the permission bits for them is given explicitly at creation time.

5.1.37 FMT_MSA.3(NI) Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the **Network Information Flow Control Policy** to provide **permissive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow

a) users with the administrator user role;

b) users with processes granted the CAP_NET_ADMIN capability and CAP_NET_RAW capability to specify alternative initial values to override the default values when an object or information is created

Application Note: The default value specified in this SFR applies to the default for the packet filter after boot. The administrator can configure alternative default values as outlined in FDP_IFF.1(NI).

Application Note: The *iptables* command uses a *netlink* interface to the kernel which requires that the caller possesses the *CAP_NET_ADMIN* capability.

5.1.38 FMT_MSA.4(PSO) Security attribute value inheritance

FMT_MSA.4.1 The TSF shall use the following rules to set the value of security attributes ***for Persistent Storage Objects*:**

a) The newly created object's owning UID is set to the effective UID of the calling subject;

b) The newly created object's owning GID is set to the effective GID of the calling subject with the following exception for file system objects: if the parent directory holding the newly created file system object is marked with the *SETGID* permission bit, the owning GID of the newly created file system object is set to the owning GID of the parent directory;

c) **The newly created object's permission bits are derived from the calling subject's *umask* value by masking out the *umask* bits from the permission bit set granting full access;**

5.1.39 FMT_MTD.1(AE) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **query, modify** the **set of audited events** to **administrator**.

Application Note: This SFR applies to FAU_SEL.1

5.1.40 FMT_MTD.1(AS) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **clear, configure the storage location, delete** the audit storage to **administrator**.

Application Note: This SFR applies to FAU_STG.1 where the directory used for storing the audit trail is configured.

Application Note: The configuration of these parameters is performed with the configuration file */etc/auditd/auditd.conf* which is writable to the administrator only.

5.1.41 FMT_MTD.1(AT) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **modify, add, delete** the

a) threshold of the audit trail when an action is performed;

b) action when the threshold is reached

to **administrator**.

Application Note: This SFR applies to FAU_STG.3.

Application Note: The configuration of these parameters is performed with the configuration file */etc/auditd/auditd.conf* which is writable to the administrator only.

5.1.42 FMT_MTD.1(AF) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **modify, add, delete** the **actions to be taken in case of audit storage failure** to **administrator**.

Application Note: This SFR applies to FAU_STG.4.

Application Note: The configuration of these parameters is performed with the configuration file */etc/auditd/auditd.conf* which is writable to the administrator only.

5.1.43 FMT_MTD.1(NI) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **query, modify, delete, change_default** the **security attributes for the rules governing the**

a) identification of *and matching of*[3] network data;

b) actions performed on the identified network data;

to **administrator**.

Application Note: This SFR applies to FDP_IFF.1(NI).

5.1.44 FMT_MTD.1(IAT) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **threshold for unsuccessful authentication attempts** to **administrator**.

Application Note: This SFR applies to FIA_AFL.1.

Application Note: The configuration of these parameters is performed with the PAM configuration files which are writable to the administrator only.

[3] "and matching of" is added compared with OSPP.

5.1.45 FMT_MTD.1(IAF) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **re-enable** the **authentication to the account subject to authentication failure** to **administrator**.

Application Note: This SFR applies to FIA_AFL.1.

Application Note: The account locking information is stored in the file */var/log/tallylog*. The locking information for an account is stored in a unique file (named after the account name) under this directory. Using the *pam_tally2.so* application which modifies this file, the account can be unlocked. The DAC permissions of that file ensure that only the administrator can write to the file.

5.1.46 FMT_MTD.1(IAU) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **initialize, modify, delete** the **user security attributes** to

a) administrator,

b) users authorized to modify their own general information

Application Note: This SFR applies to FIA_ATD.1, FIA_UAU.1, FIA_UID.1.

Application Note: The configuration of these parameters is performed with the configuration files */etc/passwd* and */etc/shadow* which are writeable to the administrator only. The general information of a user is the GECOS field in file */etc/passwd*.

5.1.47 FMT_MTD.1(SSH) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **modify** the **authentication methods provided by the OpenSSH server** to **the administrator.**

Application Note: This SFR applies to FIA_UAU.5.

Application Note: The configuration of this parameter is performed with the configuration file /etc/sshd_config which is writeable to the administrator only.

5.1.48 FMT_MTD.1 (AM-AP) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **modify, delete, clear** the **management of any TSF data** to **users allowed to invoke the application managing the TSF data or to edit the files holding the TSF data *only after another user with the role administrator has approved the action***.

Application Note: The *sudo* tool allows the root user to specify which application is executed by what user with which UID. It allows the specification of the rules fine grained down to a single application for a single user with a single target UID, including root.

5.1.49 FMT_MTD.1 (AM-MR) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **modify**, **delete** the **assignment of roles to users down to the granularity of single users** to **administrators**.

Application Note: The */etc/sudoers* file can be used to specify if each single user is granted to run some application as root, and the file is writable only to the root user.

5.1.50 FMT_MTD.1 (AM-MD) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **delegate, revoke delegation of** the **administrator user role** to **users granted that role**.

Application Note: The delegation is implemented using the *sudo* command. Every user that is allowed to use the root user can delegate parts of his responsibility by adding an appropriate rule into the */etc/sudoers* configuration file.

5.1.51 FMT_MTD.1 (AM-MA) Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to **modify, delete, clear** the **approval of administrative actions** to **administrators**.

Application Note: The */etc/sudoers* file is accessible to the root user only based on the DAC permission bits.

5.1.52 FMT_REV.1(OBJ) Revocation

FMT_REV.1.1 The TSF shall restrict the ability to revoke **object security attributes defined by SFPs** associated with the **corresponding object** under the control of the TSF to

a) (for DAC permissions): owners of the object, and administrators;

b) **(for Other security attributes): administrators.**

FMT_REV.1.2 The TSF shall enforce the ***following*** rules:

a) The access rights associated with an object shall be enforced when an access check is made;

b) no other revocation rules.

5.1.53 FMT_REV.1(USR) Revocation

FMT_REV.1.1 The TSF shall restrict the ability to revoke **user security attributes defined by the SFP** associated with the **corresponding user** under the control of the TSF to **administrators**.

FMT_REV.1.2 The TSF shall enforce the following rules:

a) The enforcement of the revocation of security-relevant authorizations with the next user-subject binding process during the next authentication of the user;

b) **No other revocation rules**

Application Note: The changes are enforced for **a new session** when the user affected by the change initiates that new session.

5.1.54 FMT_SMF.1 Specification of Management Functions

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

a) Management of auditing;

b) Management of cryptographic network protocols;

c) Management of Persistent Storage Object Access Control Policy;

d) Management of Transient Storage Object Access Control Policy;

e) Management of Network Information Flow Control Policy;

f) Management of identification and authentication policy;

g) Management of user security attributes;

h) no other management functions;

5.1.55 FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles:

a) User role with the following rights:

i. Users are authorized to modify their own user password;

ii. Users are authorized to modify the access control permissions for the named objects they own;

iii. no other rights;

b) administrator role with full privileges

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

Application Note: Administrative actions can only be performed when the calling subject possesses the above mentioned capabilities which, in the TOE configuration, is only provided to processes executing with the effective UID or file system UID of zero (also called the root user). As the account for the administrator is disabled for direct logon, authorized administrators are defined as users who are assigned to the "wheel" group. This group allows the use of the "*su*" application which is the only way to assume the root user capabilities.

5.1.56 FPT_STM.1 Reliable time stamps

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

5.1.57 FPT_TDC.1 Inter-TSF basic TSF data consistency

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret **the following TSF data types**

a) **Packet filter: protocol headers for the network protocols covered by the packet filter;** when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use **the following interpretation rules**

a) **Packet filter: protocol headers specification provided in RFC 791 (IP), RFC 793 (TCP), RFC 768 (UDP), RFC 792 (ICMP);** when interpreting the TSF data from another trusted IT product.

5.1.58 FTA_SSL.1 TSF-initiated session locking

FTA_SSL.1.1 The TSF shall lock an interactive session ***to a human user maintained by the TSF*** after **an administrator-configurable time interval of user inactivity** by:

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

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

Application Note: Clearing or overwriting the current contents of inactive session is usually applied to GUI based session interface. Since the TOE only provides command line session interface which could also be called command line shell, the related description a) is removed.

FTA_SSL.1.2 The TSF shall require the following events to occur prior to unlocking the session:

a) Successful re-authentication with the credentials of the user owning the session using password based authentication;

b) **No other events**.

Application Note: It is possible that the TSF establishes a connection to a session on a remote trusted IT system, for example when using SSH. This remote trusted IT system maintains the session established with the communication channel. The locking requirement

however applies to the session maintained by the TSF only as the TSF can only exercise control of the sessions it maintains.

5.1.59 FTA_SSL.2 User-initiated locking

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

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

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

Application Note: Clearing or overwriting the current contents of inactive session is usually applied to GUI based session interface. Since the TOE only provides command line session interface which could also be called command line shell, the related description a) is removed.

FTA_SSL.2.2 The TSF shall require the following events to occur prior to unlocking the session:

a) Successful re-authentication with the credentials of the user owning the session using password based authentication;

b) **No other events to occur**.

Application Note: It is possible that the TSF establishes a connection to a session on a remote trusted IT system, for example when using SSH. This remote trusted IT system maintains the session established with the communication channel. The locking requirement however applies to the session maintained by the TSF only, as the TSF can only exercise control of the sessions it maintains.

5.1.60 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 ***~~or~~ and*** disclosure ***using the following mechanisms:***

a) Cryptographically-protected communication channel using [
SSH protocol version 2 as defined in RFCs 4251, 4252, 4253, and 4254 with a combination of the following cipher suites defined there:

1) Symmetric ciphers defined in FCS_COP.1(NET) for encryption;

2) Keyed hash algorithms defined in FCS_COP.1(NET) for integrity;

3) Algorithms defined in FCS_CKM.2(NET) for key exchange;

*4) **Asymmetric ciphers defined in FCS_COP.1(NET) for public key encryption;***

]

b) no physically protected communication channel;

c) ***no other mechanisms for trusted communication channels***.

FTP_ITC.1.2 The TSF shall permit **the TSF, another trusted IT product** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **all security functions specified in the ST that interact with remote trusted IT systems and no other conditions or functions**.

Application Note: The SSH protocol implements a bi-directional authentication mechanism as follows:

- Server-side authentication: the user identification and authentication via user name and password / SSH user key allows the server to authenticate the client.
- Client-side authentication: the SSH host key verification performed by the

SSH client during each connection attempt allows the client to authenticate the server.

5.2 Rationale for Security Functional Requirements

This section provides the rationale for the internal consistency and completeness of the security functional requirements defined in this Security Target.

5.2.1 Security Requirements Coverage

SFR	Objectives
FAU_GEN.1	O.AUDITING
FAU_GEN.2	O.AUDITING
FAU_SAR.1	O.AUDITING
FAU_SAR.2	O.AUDITING
FAU_SEL.1	O.AUDITING
FAU_STG.1	O.AUDITING
FAU_STG.3	O.AUDITING
FAU_STG.4	O.AUDITING
FCS_CKM.1(SYM)	O.CRYPTO.NET
FCS_CKM.1(RSA)	O.CRYPTO.NET
FCS_CKM.1(DSA)	O.CRYPTO.NET
FCS_CKM.2(NET)	O.CRYPTO.NET
FCS_CKM.4	O.CRYPTO.NET
FCS_COP.1(NET)	O.CRYPTO.NET
FDP_ACC.1(PSO)	O.DISCRETIONARY.ACCESS
FDP_ACC.1(TSO)	O.SUBJECT.COM
FDP_ACF.1(PSO)	O.DISCRETIONARY.ACCESS
FDP_ACF.1(TSO)	O.SUBJECT.COM
FDP_IFC.2(NI)	O.NETWORK.FLOW
FDP_IFF.1(NI)	O.NETWORK.FLOW
FDP_ITC.2	O.DISCRETIONARY.ACCESS O.SUBJECT.COM O.NETWORK.FLOW
FDP_RIP.2	O.AUDITING O.CRYPTO.NET O.DISCRETIONARY.ACCESS O.SUBJECT.COM O.NETWORK.FLOW O.I&A

FDP_RIP.3	O.AUDITING O.CRYPTO.NET O.DISCRETIONARY.ACCESS O.SUBJECT.COM O.NETWORK.FLOW O.I&A
FIA_AFL.1	O.I&A
FIA_ATD.1(HU)	O.I&A
FIA_ATD.1(TU)	O.NETWORK.FLOW
FIA_SOS.1	O.I&A
FIA_UAU.1	O.I&A
FIA_UAU.5	O.I&A
FIA_UAU.7	O.I&A
FIA_UID.1	O.I&A O.NETWORK.FLOW
FIA_USB.2	O.I&A
FMT_MSA.1(PSO)	O.MANAGE
FMT_MSA.1(TSO)	O.MANAGE
FMT_MSA.3(PSO)	O.MANAGE
FMT_MSA.3(TSO)	O.MANAGE
FMT_MSA.3(NI)	O.MANAGE
FMT_MSA.4(PSO)	O.MANAGE
FMT_MTD.1(AE)	O.MANAGE
FMT_MTD.1(AS)	O.MANAGE
FMT_MTD.1(AT)	O.MANAGE
FMT_MTD.1(AF)	O.MANAGE
FMT_MTD.1(NI)	O.MANAGE
FMT_MTD.1(IAT)	O.MANAGE
FMT_MTD.1(IAF)	O.MANAGE
FMT_MTD.1(IAU)	O.MANAGE
FMT_MTD.1(SSH)	O.MANAGE
FMT_MTD.1(AM-AP)	O.ROLE.APPROVE
FMT_MTD.1(AM-MR)	O.ROLE.MGMT
FMT_MTD.1(AM-MD)	O.ROLE.DELEGATE
FMT_MTD.1(AM-MA)	O.ROLE.APPROVE
FMT_REV.1(OBJ)	O.MANAGE
FMT_REV.1(USR)	O.MANAGE
FMT_SMF.1	O.MANAGE
FMT_SMR.1	O.MANAGE
FPT_STM.1	O.AUDITING

FPT_TDC.1	O.DISCRETIONARY.ACCESS O.SUBJECT.COM O.NETWORK.FLOW
FTA_SSL.1	O.I&A
FTA_SSL.2	O.I&A
FTP_ITC.1	O.TRUSTED_CHANNEL

Table 14 Security Functional Requirements coverage

5.2.2 Security Requirements Sufficiency

Objectives	Coverage Rationale
O.AUDITING	The events to be audited are defined in [FAU_GEN.1] and are associated with the identity of the user that caused the event [FAU_GEN.2]. Authorized users are provided the capability to read the audit records [FAU_SAR.1], while all other users are denied access to the audit records [FAU_SAR.2]. The authorized user must have the capability to specify which audit records are generated [FAU_SEL.1]. The TOE prevents the audit log from being modified or deleted [FAU_STG.1] and ensures that the audit log is not lost due to resource shortage [FAU_STG.3, FAU_STG.4]. To support auditing, the TOE is able to maintain proper time stamps [FPT_STM.1]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].
O.CRYPTO.NET	The cryptographically-protected network protocol [FCS_COP.1(NET)] is supported by the generation of symmetric keys [FCS_CKM.1(SYM-SSH)], as well as asymmetric keys [FCS_CKM.1(RSA), FCS_CKM.1(DSA), and the functionality is supported with an internal random number generator. As part of the cryptographic network protocol, the TOE securely exchanges the symmetric key with a remote trusted IT system [FCS_CKM.2(NET)]. The TOE ensures that all keys are zeroized upon de-allocation [FCS_CKM.4]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].
O.DISCRETIONARY.ACCESS	The TSF must control access to resources based on the identity of users that are allowed to specify which resources they want to access for storing their data. The access control policy must have a defined scope of control [FDP_ACC.1(PSO)]. The rules for the access control policy are defined [FDP_ACF.1(PSO)]. When import of user data is allowed, the TOE must ensure that user data security attributes required by the access control policy are correctly interpreted [FDP_ITC.2, FPT_TDC.1]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].

O.NETWORK.FLOW	The network information flow control mechanism controls the information flowing between different entities [FDP_IFC.2(NI)]. The TOE implements a rule-set governing the information flow [FDP_IFF.1(NI)]. To facilitate the information flow control, the information must be identified [FIA_UID.1] based on security attributes the TOE can maintain [FIA_ATD.1(TU)]. The TOE must ensure that security attributes of the network data required by the information flow control policy are correctly interpreted [FDP_ITC.2, FPT_TDC.1]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].
O.SUBJECT.COM	The TSF must control the exchange of data using transient storage objects between subjects based on the identity of users. The access control policy must have a defined scope of control [FDP_ACC.1(TSO)]. The rules for the access control policy are defined [FDP_ACF.1(TSO)]. When import of user data is allowed, the TOE must ensure that user data security attributes required by the access control policy are correctly interpreted [FDP_ITC.2, FPT_TDC.1]. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3].
O.I&A	The TSF must ensure that only authorized users gain access to the TOE and its resources. Users authorized to access the TOE must use an identification and authentication process [FIA_UID.1, FIA_UAU.1]. Multiple I&A mechanisms are allowed as specified in [FIA_UAU.5]. To ensure authorized access to the TOE, authentication data is protected [FIA_ATD.1(HU), FIA_UAU.7]. Proper authorization for subjects acting on behalf of users is also ensured [FIA_USB.2]. The appropriate strength of the authentication mechanism is ensured [FIA_SOS.1]. To support the strength of authentication methods, the TOE is capable of identifying and reacting to unsuccessful authentication attempts [FIA_AFL.1]. In addition, user-initiated and TSF-initiated session locking [FTA_SSL.1, FTA_SSL.2] protect the authenticated user's session. The protection of reused resources ensures that no data leaks from other protected sources [FDP_RIP.2, FDP_RIP.3] are present.
O.MANAGE	The TOE provides management interfaces globally defined in [FMT_SMF.1] for: the access control policies [FMT_MSA.1(PSO), FMT_MSA.1(TSO), FMT_MSA.3(PSO), FMT_MSA.3(TSO)]; the information flow control policy [FMT_MSA.3(NI), FMT_MTD.1(NI)]; the auditing aspects [FMT_MTD.1(AE), FMT_MTD.1(AS), FMT_MTD.1(AT), FMT_MTD.1(AF)]; the identification and authentication aspects [FMT_MTD.1(IAT),

	FMT_MTD.1(IAF), FMT_MTD.1(IAU), FMT_MTD.1(SSH)]; Persistently stored user data is stored either in hierarchical or relational fashion, which implies an inheritance of security attributes from parent object [FMT_MSA.4(PSO)]. The rights management for the different management aspects is defined with [FMT_SMR.1]. The management interfaces for the revocation of user and object attributes is provided with [FMT_REV.1(OBJ) and FMT_REV.1(USR)].
O.TRUSTED_CHANNEL	The TOE provides a trusted channel protecting communication between a remote trusted IT system and itself [FTP_ITC.1].
O.ROLE.DELEGATE	The delegation of roles is defined and specified in [FMT_MTD.1(AM-MD)].
O.ROLE.MGMT	The definition and management of rights based on roles is defined in [FMT_MTD.1(AM-MR)].
O.ROLE.APPROVE	The approval mechanism for roles is defined with [FMT_MTD.1(AM-AP)], supported by management of the approval mechanism, i.e., specification of which roles can approve which operations [FMT_MTD.1(AM-MA)].

Table 15 Security Functional Requirements rationale

5.2.3 Security Requirements Dependency Analysis

SFR	Dependencies	Resolution
FAU_GEN.1	FPT_STM.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1
	FIA_UID.1	FIA_UID.1
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1
FAU_SEL.1	FAU_GEN.1	FAU_GEN.1
	FMT_MTD.1	FMT_MTD.1(AE)
FAU_STG.1	FAU_GEN.1	FAU_GEN.1
FAU_STG.3	FAU_STG.1	FAU_STG.1
FAU_STG.4	FAU_STG.1	FAU_STG.1
FCS_CKM.1(SYM)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.1(RSA)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.1(DSA)	[FCS_CKM.2 or FCS_COP.1]	FCS_COP.1(NET)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.2(NET)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM), FCS_CKM.1(RSA), FCS_CKM.1(DSA)
	FCS_CKM.4	FCS_CKM.4
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1(SYM)
FCS_COP.1(NET)	[FDP_ITC.1 or FDP_ITC.2 or	FCS_CKM.1(SYM),

	FCS_CKM.1]	FCS_CKM.1(RSA), FCS_CKM.1(DSA)
	FCS_CKM.4	FCS_CKM.4
FDP_ACC.1(PSO)	FDP_ACF.1	FDP_ACF.1(PSO)
FDP_ACC.1(TSO)	FDP_ACF.1	FDP_ACF.1(TSO)
FDP_ACF.1(PSO)	FDP_ACC.1	FDP_ACC.1(PSO)
	FMT_MSA.3	FMT_MSA.3(PSO)
FDP_ACF.1(TSO)	FDP_ACC.1	FDP_ACC.1(TSO)
	FMT_MSA.3	FMT_MSA.3(TSO)
FDP_IFC.2(NI)	FDP_IFF.1	FDP_IFF.1(NI)
FDP_IFF.1(NI)	FDP_IFC.1	FDP_IFC.2(NI)
	FMT_MSA.3	FMT_MSA.3(NI)
FDP_ITC.2	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(PSO) FDP_ACC.1(TSO)
	[FTP_ITC.1 or FTP_TRP.1]	FTP_ITC.1
	FPT_TDC.1	FPT_TDC.1
		FDP_IFC.2(NI)
FDP_RIP.2	No dependencies	
FDP_RIP.3	No dependencies	
FIA_AFL.1	FIA_UAU.1	FIA_UAU.1
FIA_ATD.1(HU)	No dependencies	
FIA_ATD.1(TU)	No dependencies	
FIA_SOS.1	No dependencies	
FIA_UAU.1	FIA_UID.1	FIA_UID.1
FIA_UAU.5	No dependencies	
FIA_UAU.7	FIA_UAU.1	FIA_UAU.1
FIA_UID.1	No dependencies	
FIA_USB.2	FIA_ATD.1	FIA_ATD.1(HU)
FMT_MSA.1(PSO)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(PSO)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.1(TSO)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(TSO)
	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MSA.3(PSO)	FMT_MSA.1	FMT_MSA.1(PSO)
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3(TSO)	FMT_MSA.1	FMT_MSA.1(TSO)
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.3(NI)	FMT_MSA.1	NO, but satisfied with FMT_MTD.1(NI)
	FMT_SMR.1	FMT_SMR.1
FMT_MSA.4(PSO)	[FDP_ACC.1 or FDP_IFC.1]	FDP_ACC.1(PSO)
FMT_MTD.1(AE)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AS)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AT)	FMT_SMR.1	FMT_SMR.1

	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AF)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(NI)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(IAT)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(IAF)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(IAU)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(SSH)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AM-AP)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AM-MR)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AM-MD)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_MTD.1(AM-MA)	FMT_SMR.1	FMT_SMR.1
	FMT_SMF.1	FMT_SMF.1
FMT_REV.1(OBJ)	FMT_SMR.1	FMT_SMR.1
FMT_REV.1(USR)	FMT_SMR.1	FMT_SMR.1
FMT_SMF.1	No dependencies	
FMT_SMR.1	FIA_UID.1	FIA_UID.1
FPT_STM.1	No dependencies	
FPT_TDC.1	No dependencies	
FTA_SSL.1	FIA_UAU.1	FIA_UAU.1
FTA_SSL.2	FIA_UAU.1	FIA_UAU.1
FTP_ITC.1	No dependencies	

Table 16 Security Functional Requirements dependency analysis

Rationale for unresolved dependencies:

- FMT_MSA.3(NI): FMT_MTD.1(NI) is specified to require the management of security attributes for the Network Information Flow Control Policy, just as a potential FMT_MSA.1(NI) would have been specified. However, the Network Information Flow Control Policy is not required to be enforced when managing the security attributes, as the management aspect of the network information flow control functionality is not protected by the network information flow control mechanism. Therefore, FMT_MSA.1 is not applicable and is replaced with FMT_MTD.1(NI).

5.3 Security Assurance Requirements

The security assurance requirements for the TOE include components below as specified in [CC3] part 3:

- Evaluation Assurance Level 4 components
- ALC_FLR.2

Assurance Class	Assurance Components
ADV: Development	ADV_ARC.1 Security architecture description ADV_FSP.4 Complete functional specification ADV_IMP.1 Implementation representation of the TSF ADV_TDS.3 Basic modular design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation ALC_CMS.4 Problem tracking CM coverage ALC_DEL.1 Delivery procedures ALC_DVS.1 Identification of security measures ALC_LCD.1 Developer defined life-cycle model ALC_FLR.2 Flaw reporting procedures ALC_TAT.1 Well-defined development tools
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: Test	ATE_COV.2 Analysis of coverage ATE_DPT.1 Testing: basic design ATE_FUN.1 Functional Testing ATE_IND.2 Independent Testing - sample
AVA: Vulnerability assessment	AVA_VAN.3 Focused Vulnerability analysis

No operations are applied to the assurance components apart from the operation to ASE_CCL.1 as defined in [OSPP].

5.4 Rationale for Security Assurance Requirements

The rationale for the refinement of ASE_CCL.1 is provided in [OSPP].

The basis for the justification of EAL4 requirements augmented with ALC_FLR.2 is the threat environment experienced by the typical consumers of the TOE. This matches the package description

for EAL4 (enhanced-basic).

6. TOE Summary Specification

This section explains how the security functions are implemented. The different TOE security functions cover the various SFR classes.

6.1 Audit

The Lightweight Audit Framework (LAF) is used in the audit subsystem of the TOE, which is compliant with the requirements from Common Criteria. The TOE kernel implements the core of the LAF functionality. It gathers all audit events, analyzes these events based on the audit rules, collects related information, and forwards the audit events logs that are requested to be audited to the audit daemon (*auditd*) executing in user space.

The audit functionality of the OS kernel is controlled by an audit management tool (*auditctl*(8)) in user space, which communicates with the kernel through a specific *netlink* channel. This *netlink* channel is usable only by applications with the following capabilities:

- CAP_AUDIT_CONTROL: Performing management operations like adding or deleting audit rules, setting or getting auditing parameters;
- CAP_AUDIT_WRITE: Submitting audit records to the kernel which in turn forwards the audit records to the audit daemon.

The TOE Audit security functionality includes:

- Audit event selection
- Audit trail
- Audit log overflow protection
- Audit access protection

6.1.1 Audit event selection

LAF is able to intercept all system calls and retrieve audit log entries from privileged applications (or services) in user space. The audit subsystem allows selecting the events to be actually audited from the set of all possibly auditable events based on a group of audit rules. These audit rules are set in a rule configuration file (*/etc/audit/audit.rules*), and the kernel is then notified to build its own internal structure for the events to be audited.

The audit management tool (*auditctl*(8)) is used to configure audit rules and load the audit rules from the rule configuration file. The audit rules can be modified at runtime of the system.

This security function covers the SFRs of: FAU_SEL.1.

6.1.2 Audit trail

An audit record consists of one or more lines of text in a format like “keyword=value”. The following information is contained in all audit record lines:

- Type: indicates the type of the event, such as SYSCALL, PATH, USER_LOGIN, or USER_MGMT;
- Timestamp: Date and time when the audit record was generated;
- Audit ID: unique numerical event identifier;

- Login ID ("auid"): the user ID of the user authenticated by the system (regardless if the user has changed his real and / or effective user ID afterwards);
- Effective user and group ID: the effective user and group ID of the process at the time the audit event was generated;
- Event outcome: Success or failure (where appropriate);
- Process ID of the subject that caused the event (PID);
- Hostname or terminal the subject used for performing the operation;
- Information about the intended operation;

Reliable time stamp is included in audit records. The information above is followed by event specific data. For example, for SYSCALL event records that involve file system objects, multiple text lines will be generated for a single event, all having the same time stamp and audit ID to permit easy correlation.

The audit trail is stored in ASCII text format and the administrators have the capability to read it. The TOE provides tools for managing audit trails, which can be used for post-processing of audit data. The tool, *ausearch*(8), allows selective extraction of records from the audit trail using defined selection criteria. It supports the specification of a fine-grained search pattern where each information component can be searched for, including combinations of these patterns.

This security function covers the SFRs of: FAU_GEN.1, FAU_GEN.2, FPT_STM.1, FAU_SAR.1.

6.1.3 Audit log overflow protection

When an event is to be audited according to the audit rules, the kernel would send it to the user space audit daemon (*auditd*) for storing, as an audit record, through the *netlink* channel mentioned above. The audit daemon writes the audit records to the audit trail. If the size of audit trail reaches a pre-configured warning threshold, the root user is notified about the condition and he can then backup audit trails and make room for new audit records. If the audit trail is full, the audit daemon would overwrite the oldest stored audit records, depending on the setting in the configuration file *auditd.conf*.

This security function covers the SFRs of: FAU_STG.3, FAU_STG.4.

6.1.4 Audit access protection

Access to audit data (including the configuration files and audit trails) by normal users is prohibited by the discretionary access control function of the TOE. The permission is granted only to the root user.

This security function covers the SFRs of:FAU_SAR.2, FAU_STG.1 .

6.2 Cryptographic services

The TOE offers different types of cryptographic services in different layers to protect user data:
- Cryptographic services in kernel, a socket interface is provided to user space applications to make use of this kind of the services.
- Cryptographic algorithms in several libraries for general use in user space. These libraries include *openssl*.
- Cryptographically secured network communication channels (**trusted channel**) to allow remote users to interact with the TOE. Using one type of the cryptographically secured network channels, a user can request the following services:

- OpenSSH: The OpenSSH application suite provides access to the command line interface of the TOE. OpenSSH can provide interactive as well as non-interactive sessions, and the console provided via OpenSSH provides the same environment as a local console. OpenSSH implements the SSHv2 protocol. The cryptographic primitives are provided by OpenSSL.

6.2.1 SSHv2 Protocol

The TOE provides the Secure Shell Protocol Version 2 (SSH V2) to allow users from a remote host to establish a secure connection and perform a logon to the TOE.

The following table lists implementation details concerning the OpenSSH implementation's compliance to the relevant standards. It addresses areas where the standards permit different implementation choices such as optional features.

Reference	Description	Implementation Details
RFC 4253 chapter 5	Compatibility with old SSH versions	The OpenSSH implementation is capable of interoperating with clients and servers using the old 1.x protocol. That functionality is explicitly disabled in the evaluated configuration, it permits protocol version 2.0 exclusively.
RFC 4253 section 6.2	Compression	OpenSSH supports the OPTIONAL "zlib" compression method.
RFC 4253 section 6.3	Encryption	The ciphers supported in the evaluated configuration are listed in FCS_COP.1(NET) for the SSH protocol.
RFC 4252 chapter 7	Public Key Authentication Method: "publickey"	This REQUIRED authentication method is supported by OpenSSH but can be disabled by the administrator of the OpenSSH daemon.
RFC 4252 chapter 8	Password Authentication Method: "password"	This SHOULD authentication method is supported by OpenSSH but can be disabled by the administrator of the OpenSSH daemon.
RFC 4252 chapter 8	Password change request and setting new password	The OpenSSH implementation supports the optional password change mechanism in the evaluated configuration.
RFC 4252 chapter 9	Host-Based Authentication: "hostbased"	This OPTIONAL authentication method is disabled in the evaluated configuration.

Table 17 SSH implementation notes

The TOE supports the generation of RSA and DSA key pairs. These key pairs are used by OpenSSH for the host keys as well as for the per-user keys. When a user registers his public key with the user he wants to access on the server side, a key-based authentication can be performed instead of a password-based authentication. The key generation mechanism uses the random number generator in the kernel. The evaluated configuration permits the import of externally-generated key pairs.

This security function covers the SFRs of: FCS_CKM.1(RSA), FCS_CKM.1(DSA).

The TOE supports the following security functions of the SSH V2 protocol:

- Establishing a secure communication channel using the following cryptographic functions provided by the SSH V2 protocol:

- Encryption as defined in section 6.3 of [RFC4253] - the keys are generated using the random number generator of the underlying cryptographic library;
- Diffie-Hellman key agreement as defined in section 7 of [RFC4253];
- The keyed hash function for integrity protection as defined in section 2 of [RFC6668].

Note: The protocol supports more cryptographic algorithms than the ones listed above. Those other algorithms are not covered by this evaluation and should be disabled or not used when running the evaluated configuration.

- Performing user authentication using the standard password-based authentication method the TOE provides for users (password authentication method as defined in chapter 5 of [RFC4252].
- Performing user authentication using a RSA key-based authentication method (public key authentication method as defined in chapter 5 of [RFC4252]).
- Checking the integrity of the messages exchanged and close down the connection in case an integrity error is detected.

The OpenSSH applications of sshd, ssh and ssh-keygen use the OpenSSL random number generator seeded by pulling data from */dev/random* or */dev/urandom* to generate cryptographic keys.

The cryptographic implementations ensure that sensitive data is appropriately zeroized before releasing the associated memory.

This security function covers the SFRs of: FCS_CKM.1(SYM), FCS_CKM.2(NET), FCS_CKM.4, FCS_COP.1(NET), FTP_ITC.1, FMT_SMF.1.

6.3 Packet filter

The TOE kernel's network stack implementation follows the layering structure of the network protocols. It implements the code for handling the network layer. For this layer, independent filter mechanism is provided:

- Network layer: netfilter/iptables implements the filtering mechanism for non-bridge interfaces

6.3.1 Network layer filtering

6.3.1.1 Netfilter

Netfilter is a framework for packet mangling, implemented in the RTOS V2X kernel network stack handling the network layer. The netfilter framework comprises of the following parts:

- The IP stack defines four hooks which are well-defined points in a network packet's traversal of the IP protocol stack. Each of the hooks, the network stack will call the netfilter framework allowing it to operate on the entire packet. Note: the netfilter framework provides such hooks in a number of network protocol implementations, the TOE supports TCP, UDP, IP and ICMP protocols as outlined above. Therefore, the ST specification covers TCP, UDP, IP and ICMP protocols .
- The netfilter framework provides register functions for other kernel parts to listen to the different hooks. When a packet traverses one of the hooks and passed to the netfilter framework, it invokes every registered kernel part. These kernel parts then can examine the packet and possibly alter it. As part of the examination, these kernel parts can instruct the netfilter framework to discard the packet, to allow it to pass.

The netfilter framework implements the five hooks at the following points in the packet traversal chain:

- When the packet enters the network layer of the TOE and after applying some sanity checks, but before the routing table is consulted, the NF_IP_PRE_ROUTING hook is triggered.
- After passing the routing table decision and the routing code marks the packet to be targeted for the local system, the NF_IP_LOCAL_IN hook is triggered.
- When the packet traversed the entire network stack and is about to be placed on the wire again, the NF_IP_POST_ROUTING hook is triggered.
- When a packet is generated locally, the NF_IP_LOCAL_OUT hook is triggered before the routing table is consulted.

6.3.1.2 Iptables

All communication on the network layer can be controlled by the *IPTables* framework. The TOE implements a packet filter as part of the network stack provided with the kernel. The combination of *IPTables* and *netfilter* implements the packet filter which provides stateful and stateless packet filtering for network communication by inspecting the IP header, the TCP header, UDP header and/or ICMP header of every network packet that passes the network stack.

The packet selection system called *IP Tables* uses the netfilter framework to implement the actual packet filtering logic on the network layer for the TCP/IP protocol family.

IPTables consists of the following two components:

- In-kernel packet filter enforcement: The kernel-side of IPTables use the netfilter framework as indicated above. Three lists of packet filter rules are enforced by the kernel mechanism: one for each netfilter framework hook that applies to packet filtering. When a packet is analyzed by the IPTables kernel modules, they first select the applicable list based on the hook where the netfilter framework triggered IPTables. Each list contains zero or more rules which are iterated sequentially. A rule consists of a matching part (also called the "match extension") and an action part (also called the "target extension"). When a rule is applied to a packet, the kernel modules first apply the matching part of the rule. If the packet matches, the action part is enforced. If the action part contains a decision of the fate of the packet (to accept it, to drop it, or to drop it and sending a notification to the sender), the rule list validation stops for this packet. If the action part contains a modification instruction or log instruction for the packet, the rule list validation continues after performing this operation. When the rule list is iterated through and a packet could not be matched by a rule with a decision action (accept, drop), the default decision action applicable to the list is enforced. This default action is either to accept the packet, to drop the packet, or to drop the packet and send a notification to the sender.
- User space configuration application: The user space application *iptables(1)* allows the configuration of the IPTables kernel components. The application allows the specification of one rule per invocation where a rule contains the above mentioned matching part and action part. The tool also allows modification or deletion of existing rules as well as configuration of the default action. When using the tool, each invocation must specify the netfilter framework hook to which the rule applies to. See the man page of *iptables(1)* for more details.

This security function covers the SFRs of:

- Packet filtering rules: FDP_IFC.2(NI), FDP_IFF.1(NI);
- Interpretation of network protocol: FIA_UID.1, FDP_ITC.2, FPT_TDC.1;
- Maintenance of rules: FIA_ATD.1(TU);

6.4 Identification and Authentication

User identification and authentication in the TOE includes all forms of interactive login (e.g. remote login using the SSH protocol or local login at the local console) as well as identity changes through the *su* commands. These all rely on explicit authentication information provided interactively by a user. In addition, the key-based authentication mechanism of the OpenSSH server is another form of authentication.

6.4.1 PAM-based identification and authentication mechanisms

The TOE uses a suite of libraries called the "Pluggable Authentication Modules" (PAM) that allow an administrative user to choose how PAM-aware applications authenticate users. The TOE provides PAM modules that implement all security functionalities listed:

- Providing login control and establishing UIDs, GIDs and login ID for a subject;
- Ensuring the quality of passwords, and ensuring the minimum strength used for authentication(passwords);
- Enforcing limits for accounts (such as the number of maximum concurrent sessions allowed for a user);
- Enforcing the change of passwords after a configured time including the password quality enforcement;
- Enforcement on session locking of accounts after exceeding maximum failed login attempts;
- Restriction on the use of root account to certain terminals;
- Restriction on the use of the *su* commands

Following attributes are set for the user at login time:
- the real, effective, file system UID;
- the real, effective, file system GID;
- the set of supplementary GIDs of the subject that is created;
- the login UID;

It is up to the client application, which is usually provided by a remote system, to protect the user's entry of a password correctly (e. g. provide only obscured feedback).

During the login process, the user is shown with a banner. The login time is recorded after successfully logging in.

When configuring the OpenSSH server, the administrator is permitted to enable SSH key-based authentication or username/password based authentication. If both are enabled, the key-based authentication has the highest priority. When a user can successfully authenticate themselves using the SSH key-based authentication based on a private SSH key, the TOE grants access to the user.

After a successful identification and authentication, the TOE initiates a session for the user and spawns the initial login shell as the first process the user can interact with. The TOE provides a mechanism to lock a session either automatically after a configurable period of inactivity for that session, or upon the user's request.

User accounts are stored in configuration files (/etc/passwd and /etc/shadow). Both files are writable to the root user only. In addition, /etc/shadow is readable to the root user only. Modification of both files is performed using a set of administrative applications. When a user ID is removed, its entry in the configuration files is removed by the administrative interfaces, and all the entries of working directories and files are removed. Therefore, a login using a removed user ID will be unsuccessful.

The administrator is able to define rules and restrictions for passwords used to authenticate users. All users except those only have the "user" role are also allowed to change their own password using the

passwd command. The password restrictions defined by the administrator apply. Permissions can be changed by the object owner and an administrative user. When an object is created, the creator is the object owner. Object ownership can be transferred. In the case of IPC objects, the creator will have the same right as the owner, even when the ownership has been transferred.

The complexity of the password is set by pam-based configuration files in /etc/pam.d/*. The TOE password change is implemented using the PAM library. The PAM module pam_cracklib.so allows the specification of the quality of new passwords. The evaluated configuration requires a configuration of the PAM-based password change mechanism that meets the password complexity. Only administrator can modify the complexity of user account password.

Only administrators are able to revoke users' group membership by using command *delgroup* and remove users by using command *userdel*. Once a user has been removed, the user cannot login via SSH. The changes are enforced for a new session when the user affected by the change initiates that new session.

This security function covers the SFRs of FDP_RIP.3, FIA_AFL.1, FIA_SOS.1, FIA_UAU.1, FIA_UID.1, FIA_UAU.5, FIA_UAU.7, FIA_USB.2, FMT_REV.1(USR).

6.4.2 User Identity changing

Users can change their identity (i.e., switch to another identity) using commands *su* provided with the TOE.

6.4.2.1 su command

The *su* command is intended for a switch to another identity that establishes a new login session and spawns a new shell. When invoking *su*, the user must provide credentials associated with the target identity, i.e., when the user wants to switch to another user ID, it has to provide the password protecting the account of the target user.

The primary use of the *su* command within the TOE is to allow appropriately authorized individuals the ability to assume the root identity to perform administrative actions. In addition, the use of the *su* command to switch to root has been restricted to users belonging to a special group (*wheel*). Users that don't have access to a terminal where root login is allowed and are not member of that special group will not be able to switch their real, effective, and file system user ID to root even if they would know the authentication information for root. Note that when a user executes a program that has the *setuid* bit set, only the effective user ID and file system ID are changed to that of the owner of the file containing the program while the real user ID remains that of the caller. The login ID is neither changed by the *su* command nor by executing a program that has the *setuid* or *setgid* bit set as it is used for auditing purposes.

6.4.2.2 Changed identities

When switching identities using *su*, the real, file system and effective user ID and real, file system and effective group ID are changed to the one of the user specified in the command (after successful authentication as this user).

The *su* command invokes the common authentication mechanism to validate the supplied authentication.

This security function covers the SFRs of FIA_USB.2.

6.4.3 Authentication Data Management

Each TOE instance maintains its own set of users with their passwords and attributes. Although the same human user may have accounts on different servers interconnected by a network and running an instantiation of the TOE, those accounts and their parameter are not synchronized on different TOE instances. As a result, the same user may have different user names, different user Ids, different passwords and different attributes on different machines within the networked environment. Existing mechanism for synchronizing this within the whole networked system are not subject to this evaluation.

Each TOE instance within the network maintains its own administrative database by making all administrative changes on the local TOE instance. System administration has to ensure that all machines within the network are configured in accordance with the requirements defined in this Security Target.

The TOE maintains two security roles, the administrator and normal users, which identified by user login uid, and maintained in */etc/passwd*. User from remote trusted IT system uses *ssh-keygen(1)* to create SSH key pair, and send public key to the TOE, then the TOE can use the public key as software token to authenticate the user.

The file */etc/passwd* contains following information for each user:

- the user's name;
- the login UID (an integer) of the user;
- an indicator whether the password of the user is valid;
- the principal login GID (an integer) of the user;
- and other (not security relevant) information.

The file */etc/shadow* contains following information for each user:

- the user's name;
- a hash of the user's password ('*' for disabled account);
- the time the password was last changed;
- the expiration time of the password ('0' for changing at any time);
- the validity period of the password;
- time to warn user of an expiring password (7 for a full week);
- and some other information that are not subject to the security functions as defined in this Security Target.

Users are allowed to change their passwords by using the *passwd(1)* command. This application is able to read and modify the contents of file */etc/shadow* for the user's password entry, which would ordinarily be inaccessible to a non-privileged user process. Users are also warned to change their passwords at login time if the password will expire soon, and are prevented from logging in if the password has expired.

The TOE displays informative banners before or during the login to users. The banners can be specified in file */etc/motd* for logins via the physical console and remote logins, such as via SSH. When performing a login on the physical console, the banner is displayed above the username and password prompt. For logins via SSH, the banner is displayed to the remote peer during the SSH-session handshake takes place. The remote SSH client will display the banner to the user. When using the provided OpenSSH client, the banner is displayed when the user instructs the OpenSSH client to log into the remote system.

This security function covers the SFRs of FIA_ATD.1(HU).

6.4.4 SSH key-based authentication

In addition to the PAM-based authentication outlined above, the OpenSSH server is able to perform a key-based authentication. When a user wants to log on, instead of providing a password, the user applies his SSH key. After a successful verification, the OpenSSH server considers the user as authenticated and performs the PAM-based operations as outlined above.

To establish a key-based authentication, a user first has to generate an RSA key pair. The private part of the key pair remains on the client side. The public part is copied to the server into the file *.ssh/authorized_keys* which resides in the home directory of the user he wants to log on as. When the login operation is performed, the SSHv2 protocol tries to perform the "publickey" authentication, using the private key on the client side and the public key found on the server side. The process for the publickey authentication is defined in [RFC4252] chapter 7.

Users have to protect their private key part the same way as protecting a password. Appropriate permission settings on the file holding the private key is necessary. To strengthen the protection of the private key, the user can encrypt the key where a password serves as key for the encryption operation. See *ssh-keygen(1)* on the TOE for more information. The SSH V2 protocol applies in *sshd_config* and *ssh_config* to configure the login methods.

This security function covers the SFRs of FIA_SOS.1, FIA_UAU.1, FIA_UID.1, FIA_UAU.5, FMT_SMF.1.

6.4.5 Session locking

The TOE uses the *vlock(1)* application which locks the current session of the user either after an administrator-specified time of inactivity or upon the user's request. To unlock the session, the user must supply his password. Screen uses PAM to validate the password and allows the user to access his session after a successful validation.

This security function covers the SFRs of FTA_SSL.1, FTA_SSL.2.

6.5 Discretionary Access control

Several discretionary access control (DAC) mechanisms are implemented in RTOS V2X, however, only the standard DAC mechanism in traditional Unix systems, i.e. permission-bits and system-services-bits are covered in this evaluation. The general policy for DAC is that subjects (i.e., processes) are allowed only the accesses specified by the policies applicable to the object the subject requests access to. Further, the ability to propagate access permissions is limited to those subjects who have that permission, as determined by the policies applicable to the object the subject requests access to.

A subject may possess one or more of the following capabilities which provide the following exemptions from the DAC mechanism:

- CAP_DAC_OVERRIDE: A process with this capability is exempt from all restrictions of the discretionary access control and can perform any action desired. For the execution of a file, the permission bit vector of that file must contain at least one execute bit.
- CAP_DAC_READ_SEARCH: A process with this capability overrides all DAC restrictions regarding read and search on files and directories.
- CAP_ CHOWN: A process with this capability is allowed to make arbitrary changes to a file's UID or GID.

- CAP_FOWNER: Setting permissions and ownership on objects even if the process' UID does not match the UID of the object.
- CAP_FSETID: Don't clear SUID and SGID permission bits when a file is modified.

DAC provides the mechanism that allows users to specify and control access to objects that they own. DAC attributes are assigned to objects at creation time and remain in effect until the object is destroyed or the object attributes are changed. DAC attributes exist for, and are particular to, each type of named object known to the TOE (i.e. file system objects and IPC objects). DAC is implemented with permission bits.

The outlined DAC mechanism applies only to named objects which can be used to store or transmit user data. Other named objects are also covered by the DAC mechanism but may be supplemented by further restrictions. These additional restrictions are out of scope for this evaluation. Examples of objects which are accessible to users that cannot be used to store or transmit user data are:

- Virtual file systems externalizing kernel data structures (such as most of procfs, sysfs, binfmt_misc);
- Process signals.

During creation of objects, the TSF ensures that all residual contents are removed from that object before making it accessible to the subject requesting the creation.

When data is imported into the TOE (such as when mounting disks created by other trusted systems), the TOE enforces the permission bits (and other access control policies) applied to the file system objects.

6.5.1 Permission bits

The DAC mechanism covered in this evaluation is the standard UNIX permission bits for file system objects in all supported file systems, objects of native classes, and system-services-bit for objects of native classes. There are three sets of three-bit tuple that define access for three categories of users: 1) the owning user, 2) users in the owning group, and 3) other users. The three-bit tuple in each set indicate the access permissions granted to each user category: one bit for read (r), one for write (w) and one for execute (x). Note that write access to file systems mounted as read only is always rejected (the exceptions are character and block device files which can still be written to as write operations do not modify the information on the storage media). The “restricted deletion” (or sticky) attribute is used for world-writable temp directories, preventing the removal of files by users other than the owner. Any subjects which are not the system services cannot access an object with system-services-bit, even if the accesses are allowed by the standard DAC mechanism above.

Each process has an inheritable “umask” attribute which is used to determine the default access permissions for new objects. It is a bit mask of the user/group/other read/write/execute bits, and specifies the access bits to be removed from new objects. For example, setting the umask to “002” ensures that new objects will be writable by the owner and group, but not by others. The umask is defined by the root user in the */etc/login.defs* file or “022” by default if not specified. The “unmask” does not control the system-services-bit.

Only processes which are system services can create objects with system-services-bit and set or clear the system-services-bit of objects.

This security function covers the SFRs of FDP_ACC.1(PSO), FDP_ACF.1(PSO), FDP_ITC.2, FDP_RIP.2.

6.5.2 File system objects

Access to file system objects is generally governed by permission bits. File system objects access checks are performed when the object is initially opened, and are not checked on each subsequent access. Changes to access controls (i.e., revocation) are effective with the next attempt to open the object.

6.5.3 IPC objects

The TOE implements the following standard types of IPC mechanisms:

- SysV Shared Memory
- SysV and POSIX Message Queues
- SysV Semaphores

  And the following native types IPC mechanisms:
- Native Shared Memory
- Native Message Queues
- Native Path

Access to the above mentioned IPC mechanisms are governed by UNIX permission bits. Native IPC mechanisms are governed by the system-services-bit.

Two other types of IPC objects, UNIX domain socket special files and Named Pipes, are represented as file system objects, and, the access control mechanism covering file system objects are applicable to these IPC mechanisms too.

The TOE maintains another IPC object type where each process has its own namespace for that: sockets (including network sockets). Access to the socket is only possible by the process whose socket namespace contains the socket reference. Setting of permissions for such objects can be handled using file descriptor passing.

Modification of DAC attributes is restricted to the owner of the object or users with the aforementioned capabilities.

This security function covers the SFRs of FDP_ACC.1(TSO), FDP_ACF.1(TSO), FMT_REV.1(OBJ).

6.6 Security Management

The security management facilities provided by the TOE are usable by authorized users and/or authorized administrators to modify the configurations of TSF. The configurations of TSF are hosted in the following locations:

- Configuration files (or TSF databases);
- Data structures maintained by the kernel and within the kernel memory;

The TOE provides applications to authorized users as well as authorized administrators to perform various administrative tasks. These applications are documented as part of the administrator and user guidance. These applications are either used to modify configuration files or to access parameters controlled and enforced by the kernel via kernel-provided interfaces to user space.

Configuration options are stored in different configuration files. These files are protected using the DAC mechanisms against unauthorized access where usually the root user only is allowed to write to the files. In some special cases (like for /etc/shadow), the file is even readable to the root user only. The TOE provides *chmod(1)* to set the file to the correct permission bit, and provides *chown(1),*

chgrp(1) to change file owner and group. And the default permission bit is derived from the *umask* value, the default global *umask* value is set in the */etc/login.defs*. Users can change their umask value at any time using the *umask(2)* system call. It is the task of the persons responsible for setting up and administrating the system to ensure that the access control features of the TOE are used throughout the lifetime of the system to protect those databases. These configuration files are accessed using applications which are able to interpret the contents of these configuration files. Each TOE instance maintains its own TSF database. Synchronizing those databases is not performed in the evaluated configuration. If such synchronization is required by an organization, it is the responsibility of an administrative user of the TOE to achieve this, either manually or with some automated assistance.

To access data structures maintained by the kernel, applications use the kernel-provided interfaces, such as:

- system calls;
- system interfaces;
- virtual file systems;
- netlink sockets;
- and device files.

These kernel interfaces are restricted to authorized administrators or authorized users, if applicable, by either using DAC (for virtual file system objects) or special kernel-internal verification checks for each interface.

The TOE provides security management applications for all security-relevant settings listed throughout this ST, i.e.:

- DAC mechanism chmod(1), chown(1), chgrp(1) to set file permission bit and ownership, and hm_mem_shm_chmod, hm_path_chmod, hm_mqueue_chmod to set permission bit and system-services-bit for native shared memories, paths and message queues;
- *sudo* command used to grant user running some applications as root;
- *iptables* command to control the network packet filter rule;
- *pam_tally2.so* to lock the user who exceeds the threshold of unsuccessful authentication attempts, only administrator can modify /etc/pam.d/* configuration files to change the threshold for unsuccessful attempts and use command *pam_tally2* to re-enable the authentication to account subject to authentication failures;

The configuration files in /etc/pam.d manage I&A policy.

The administrator can manage user security attributes, which stored in /etc/passwd, /etc/group and /etc/shadow. The user can change his own passwords.

The TOE provides security management applications for all security-relevant settings listed throughout this ST,.The security management covers all FMT_MSA.1(PSO), FMT_MSA.1(TSO), FMT_MSA.3(PSO), FMT_MSA.3(TSO), FMT_MSA.3(NI), FMT_MSA.4(PSO), FMT_MTD.1(AE), FMT_MTD.1(AS), FMT_MTD.1(AT), FMT_MTD.1(AF), FMT_MTD.1(NI), FMT_MTD.1(IAT), FMT_MTD.1(IAF), FMT_MTD.1(IAU), FMT_MTD.1(SSH), FMT_SMR.1, FMT_SMF.1.

6.7 Privileges

The TOE maintains two roles, the administrator and normal users. By default, only the root user is in the administrator role, and can perform any operations on the system. Normal users can only operate on objects they own or they are granted by the administrator to operate on. The administrator can also delegate his privilege to specified normal users after successful authentication.

Privileges to perform administrative actions are maintained by the TOE. These privileges are separated into privileges to act on data or access functionality in user space and in kernel space.

Functionality accessible in user space are applications that can be invoked by users. Also, data accessible in user space is either data maintained with an application or data stored in persistent or transient storage objects. Privileges are controlled by permissions to invoke applications and to access data. For example, the configuration files including the user databases of */etc/passwd* is writable only to the root user and accessible to all users, and */etc/shadow* are accessible and readable to the root user only. Therefore, the root user is given the privilege to perform modifications on this configuration data, which constitutes administrative actions.

Functionality and data maintained by the kernel must be accessed using system calls. The kernel implements a privilege check for functions and data that shall not be accessible by normal users. These privileges are controlled with capabilities that can be assigned to processes. If a process is assigned with a capability, it is allowed to request special operations that other processes cannot. To implement consistency with the Unix legacy, processes with the effective UID of zero are implicitly given all capabilities. However, these processes may decide to drop capabilities. Such capabilities are marked by names with the prefix of "CAP_" throughout this document. The RTOS_V2X kernel implements many more capabilities than mentioned in this document. These unmentioned capabilities protect functions that do not directly cover SFR functionality but need to be protected to ensure the integrity of the system and its resources.

This security function covers all SFRs of FMT_MTD.1(AM-AP), FMT_MTD.1(AM-MR), FMT_MTD.1(AM-MD), FMT_MTD.1(AM-MA), FMT_SMR.1, FMT_REV.1(OBJ), FMT_REV.1(USR).

7. Abbreviations, Terminology and References

7.1 Abbreviations

Abbreviation	Description
AH	Authentication Header
CC	Common Criteria
DAC	Discretionary Access Control
EAL	Evaluation Assurance Level
HMAC	Hash-based Message Authentication Code
IPA	Identity, Policy and Audit. An identity management system.
IPC	Inter Process Communication
OSPP	Operating System Protection Profile
PP	Protection Profile
RNG	Random number generator.
NPTRNG	Non-physical true random number generators
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TSF	TOE security function
TSFI	TSF Interface
TSP	TOE security policy
UID	User identifier, is a number assigned by RTOS V2X to each user on the system
GID	Group Identifier

7.2 Terminology

Owning UID	The UID of the user who owns the object.
Owning GID	The GID of the owning group of the object.
Symbolic link	It is consists of a special type of file that serves as a reference to another file or directory
Real UID	Real User Identifier (ruid), it identifies the real user identifier of the process.
Real GID	Real Group Identifier (rgid), it identifies the real group identifier of the process.
Effective UID	Effective User Identifier (euid). Once euid is set, it replaces ruid. The identifier of the process depends on euid. However, euid cannot used for access control to objects of file system, only fsuid can control it.
Effective GID	Effective Group Identifier (egid). Once egid is set, it replaces Real GID. The identifier of the process depends on egid. However, egid cannot used for access control to objects in file system, only File system group ID can control it.
File system UID	File system user ID (fsuid), once fsuid is set, it replace ruid, which is used explicitly for access control to the object of file system.
File system GID	File system group ID (fsgid), once fsgid is set, it replace rgid, which is used explicitly for access control to the object of file system.

Mount	Mount a filesystem

7.3 References

CC Common Criteria for Information Technology Security Evaluation
Version 3.1R5
Date April 2017
Location https://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf
Location https://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf
Location https://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf

FIPS180-4 Secure Hash Standard
Date August 2015
Location http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

OSPP BSI Operating System Protection Profile
Version 2.0
Date 2010

OSPP-AM OSPP Extended Package - Advanced Management
Version 2.0
Date 2010

RFC4252 The Secure Shell (SSH) Authentication Protocol
Date January 2006
Location https://datatracker.ietf.org/doc/html/rfc4252

RFC4253 The Secure Shell (SSH) Transport Layer Protocol
Date January 2006
Location http://tools.ietf.org/html/rfc4253

RFC4344 The Secure Shell (SSH) Transport Layer Encryption Modes
Date January 2006
Location https://tools.ietf.org/html/rfc4344

RFC4419 Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol
Date March 2006
Location https://tools.ietf.org/html/rfc4419

RFC5647 AES Galois Counter Mode for the Secure Shell Transport Layer Protocol
Date August 2009
Location https://tools.ietf.org/html/rfc5647

RFC5656 Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer
Date December 2009
Location https://tools.ietf.org/html/rfc5656

RFC6668 SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol
Date 2013-07-01
Location https://tools.ietf.org/html/rfc6668

RFC8268 More Modular Exponentiation (MODP) Diffie-Hellman (DH) Key Exchange (KEX) Groups for Secure Shell (SSH)
Date December 2017
Location https://tools.ietf.org/html/rfc8268

RFC8332 Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol
Date March 2018
Location https://tools.ietf.org/html/rfc8332

OpenSSH daemon configuration file

manpage of ssh_config	Date	2020-04-17
	Location	https://man7.org/linux/man-pages/man5/sshd_config.5.html