VIRTUAL AIR GAP

(VAG)

v1.0.6

Security Target Lite

Version 1.10

August 2012

TABLE OF CONTENTS

1. INTRODUCTION

This section provides an introduction to the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, and the ST organization.

The TOE is Virtual Air Gap (VAG) v1.0.6 product, which is jointly designed, developed and provided by Aselsan Inc., and Invicta R&D Ltd.

This section of the document;

- identifies the Security Target (ST) and Target of Evaluation (TOE),
- specifies the ST conventions,
- provides an overview and the description of TOE,
- describes the ST organization.

This document is a sanitized version of the Security Target (ST) used for the evaluation of VAG. It is classified as public information.

1.1 ST Reference and TOE Reference

ST Title:	Virtual Air Gap (VAG) v1.0.6 Security Target Lite
ST Version:	v1.10
ST Release Date:	06 August 2012
TOE Identification:	Virtual Air Gap (VAG) v1.0.6
CC Identification:	Common Criteria for Information Technology Security Evaluations, Version 3.1R3
Keywords:	Virtual Air Gap, VAG, Border Security, Network Isolation and Separation

1.2 Conventions, Terminology & Acronyms

This section specifies the conventions and formatting information used throughout the whole ST Document.

1.2.1 Acronyms

TOE	Target of Evaluation
ST	Security Target
TSP	TOE Security Policies

VAG	Virtual Air Gap
FW	Firewall
NIDS	Network Intrusion Detection System
HIDS	Host-based Intrusion Detection System

1.2.2 Conventions

In this Security Target some notations and conventions, which are taken from the Common Criteria v3.1R3 have been used in order to guide the reader.

During the specification of the functional requirements under the Section 4, the functional components are interpreted according to the “assignment” and “selection” operations.

- The outcome of the assignment operations is shown with **bold** and identified between “[**brackets**]”.
- The outcome of the selection operations is shown with **bold** and **underlined** and identified between “[**brackets**]”.
- The iterated components are shown **ComponentID/"IterationLabel".**

1.2.3 Terminology

The following terminology is used in this Security Target (ST).

Access Control: Security service that controls the usage of assets defined in the TOE.

Management Interface (MI): WEB interface provided by vag-int for administrative users.

Management Console: Host connected to the internal network used by administrative users to access to the Management Interface.

Administrative User: Users that are granted authorization to configure and control the TOE via management interface.

Administrative User Security Attributes: TOE data associated with administrative users that is used for security policies of TOE.

Linux Shell: Command line terminal belonging to the Linux Operating System of both parts (**vag-int** and **vag-ext**) where the TOE runs. This terminal is only accessible at the physical location where both parts are.

Maintenance User: A special user having certain limited set of administrative capabilities for configuring and controlling the TOE through the Linux Shell, right after a successful login to the system though the text console via user name “consolemaintenance” and its associated password (Note: This user is not allowed to login through the management interface). This is the only user allowed to access to the physical location where **vag-int** and **vag-ext** are deployed.

VAG User: An internal/external entity that sends requests to and gets responses from (i.e., interacts with) the TOE.

System Information: vag-int status, vag-ext status, bandwidth information, liveliness between vag-int and vag-ext, active user session in the management interface, firewall status, IDS status and network status.

Configuration Data: Ethernet interface definitions of internal and external hosts, IDS status, firewall rules, web and mail.

Full backup: A system backup of either **vag-int** or **vag-ext** containing disks configuration information, network configuration information, postfix configuration information, firewall configuration information, proxy configuration information, users information, audit information and alert information.

Partial backup: A backup containing Ethernet interface definitions and firewall information.

Cryptographic keys: Secret data used for cryptographic operations whose original copies are kept on a USB Flash Disk (Token).

Cryptographic operation: A cryptographic algorithm's action that perform any of the following operations;

- transforms plaintext into ciphertext.
- Transforms ciphertext into plaintext.
- a digital signature computed from data.
- verification of a digital signature

Invictus: Whole system, which consists of server hardware, operating system, TOE, and external security means such as Firewall, NIDS, and HIDS.

Internal Host (vag-int): Invictus that is deployed to the internal network, which contains a management interface.

External Host (vag-ext): Invictus that is deployed to the external network and has no management interface.

Firewall: Rule based iptables software that is configured via management interface to mediate the information flow.

Host-based Intrusion Detection System (HIDS): Running software to detect intrusion to host operating system according to non-configurable pre-defined patterns.

Network Intrusion Detection System (NIDS): Running software to detect intrusion to internal or external host according to semi-configurable pre-defined patterns.

Shared disk: Shared disk array connected to **vag-int** and **vag-ext** that is the unique path for the information flow between them.

Host disk: Internal disk belonging to each part, **vag-int** and **vag-ext**.

Alarm: A system message that is displayed via management interface, indicating an unusual activity. It is a trail that matches with predefined exception patterns in the log or audit file.

Passive mode: A TOE mode of operation where the information flow between **vag-int** and **vag-ext** is disabled for VAG Users and the unique information flow between **vag-int** and **vag-ext** is for management purposes for the administrative users. The management interface and Linux Shell remain enabled.

Critical Alarm: A special type of alarm that will trigger the system to go into non-operational (passive) mode.

VAG Logs: Files stored out of the TOE (in shared disk for **`vag-ext`** and in host disk for **`vag-int`**) where are recorded the activities performed by the TOE.

System Logs: Files stored out of the TOE (in shared disk for **`vag-ext`** and in host disk for **`vag-int`**) to maintain the activities of the Linux OS that TOE runs in conjunction with.

AP: Application Protocol.

APDU: Application Protocol Data Unit.

Supported APs: Those APs that are supported by the TOE. These APs are Hyper-Text Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP).

Data Flow: Any traffic attempting to flow through the TOE.

Data input: Data flow coming from the net connected to **`vag-ext`** to the network connected to **`vag-int`**.

Data output: Data flow coming from the net connected to **`vag-int`** to the network connected to **`vag-ext`**.

Approved Data Flow: Any traffic flowing over the TOE that is not rejected due to any reason.

Rejected Data Flow: Any traffic that is filtered out as a result of its identification as malicious data or not allowed data.

1.3 TOE Overview

1.3.1 TOE Usage

The TOE, namely the Virtual Air Gap (VAG), is a software package which provides a secure network traffic flow between private and public networks in order to realize mission-critical operations fundamentally by preventing transit IP traffic. The TOE is running on internal and external host machines (**`vag-int`** and **`vag-ext`**) on top of Linux operating systems and mediates the information flow with the support of external software installed in its environment.

TOE is designed for institutions (public and private) that are connected to Internet and offering/getting real-time web and mail service and data interaction over Internet to prevent and remove security threats towards mission-critical operations.

TOE system is deployed between external network and institution’s internal network and does not use IP-based communication for internal connection. Therefore, the TOE is actually forming a “virtual air gap” border providing high-level security.

The system that runs the TOE is basically composed of internal and external security components (servers) and a shared memory (shared disk) component. Figure-1 shows the general architectural view of the TOE and its environment.

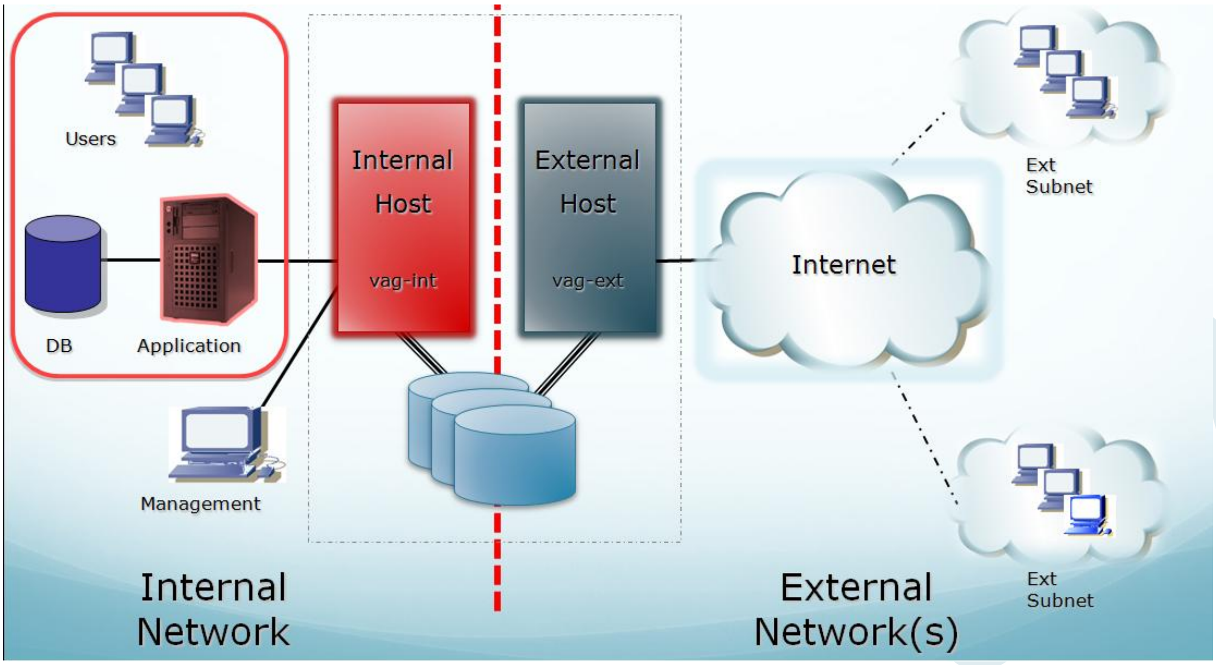


Figure 1. General Architecture of Virtual Air Gap and the Operational Environment

TOE is protected by a number of environmental components in order to function appropriately. These components include firewall (FW), network-based intrusion detection system (NIDS), protocol filter and host based intrusion detection system (HIDS) working on both servers (**vag-int** and **vag-ext**). **vag-int** has a management interface that enables administrative users (with sufficient access rights) to manage and monitor both internal and external hosts' system information, configuration data, partial backups, administrative users, audit logs and user passwords.

The TOE performs user identification and authentication and an access control policy for the administrative users. The identification and authentication of administrative users makes use of the user name and password. This password must follow a certain policy.

Information flow over TOE is bi-directional; through external to internal network and vice versa. External network's requests/responses are taken by external host (**vag-ext**). The requests/responses are passed through application level controls by a process running on external host. In case of proper validation, the requests/responses are recorded and transferred to the respective application on the internal network. Same information flow is valid for connections from internal network to external network.

The communication between **vag-int** and **vag-ext** is encrypted and signed. Crypto/Sign layer of the VAG architecture that is shown in Figure 2 invokes two cryptographic actions on the data packets flowing from message layer to disk access layer. Operational Environment first encrypts the payload of the data packet, and then signs the whole packet through crypto/sign module of the TOE. This way, disk has signed and encrypted data packets, which can only be resolved by peer host.

The shared disk array hosts a file system, which is used as a database for **vag-ext** log files. The shared disk array and the file system are not in the scope of the TOE and are considered as environmental components.

All the problems that may arise in any of these stages are recorded in the audit log; these records can be used to analyze the security or operation of the system. All the history of interactions is accessible through management interface. An automatic procedure searches the audit logs for predefined attack patterns and generates alarms in case of detecting a potential attack. The TOE is able to take certain actions in case of reaching this circumstance.

1.3.2 TOE Type

TOE is a software solution which provides data exchange between networks which have different security levels.

1.3.3 Required non-TOE hardware/software/firmware

#	Requirements	Descriptions	Version & Specifications
1	Invictus Hardware	▪ Internal and external host PCs ▪ Storage ▪ RAM ▪ Processor ▪ Monitor	There are two identical servers. Each will have (as recommended minimum); ▪ 2 x 80 GB SATA/SAS (Hardware Supported RAID 0/1) Disk ▪ 4 GB Memory ▪ Intel Dual/Quad Core Processor ▪ Acceptable resolution view based on monitor.
2	Management Console Hardware	▪ Host	▪ Any PC with a running OS supporting a web browser.
3	Management Console Software	▪ Operating System ▪ Web Browser	▪ Any OS supporting any of the following Web Browsers: ○ IE 6 and above ○ Firefox 2.5 and above ○ Chrome 5 and above ○ Opera 10 and above
4	Shared Disk Hardware	▪ Disk Array	▪ Dual port SAS or Fiber Channel Interface ▪ RAID 0/1/3/5 Support ▪ 2 GB cache ▪ 12 x 80 GB Disk Units

#	Requirements	Descriptions	Version & Specifications
5	Invictus Software	▪ OS ▪ Firewall ▪ IDS ▪ HIDS ▪ Protocol Layer (HTTP & SMTP proxy)	▪ Debian 5.0 and above ▪ Kernel 2.6.32-30~bpo50+1 ▪ iptables 1.4.2-6 ▪ Snort 2.7.0-21 ▪ Samhain 2.2.3-7 ▪ Lighttpd 1.4.25-3 ▪ Apache 2.2.17 (includes mod-proxy) ▪ mod-security 2.5.13 ▪ Postfix 2.5.5-1.1 ▪ Amavisd-new 1:2.6.1.dfsg-1 ▪ Clamav 0.97.3+dfsg-1~lenny1 ▪ Spamassasin 3.2.5-2+lenny3

1.4 TOE Description

VAG is designed for institutions (public and private) that are connected to Internet and offering/getting real-time web and mail service over Internet to prevent and remove security threats towards mission-critical operations.

1.4.1 Logical Scope

Management and Maintenance (MM)

Management interface of the TOE is the WEB interface where the administrative users can monitor and/or configure some components of the system.

Management interface is provided by internal host (**`vag-int`**) and accessed via management console that communicates over HTTPS protocol on **`vag-int`**'s network interface. Management console client, which is not in the scope of TOE, is a simple web browser application that can run JavaScript code.

The TOE performs an authentication and identification of administrative users by using the username and password. After that, an access control policy is exercised in order to provide access rights to the certain administrative user.

Administrative Users of the system are categorized into 3 groups:

- Operator
- Manager
- Administrator

In addition to the Administrative Users who access the system through a web interface, there is also a Maintenance User whose username is *"consolemaintenance"* who is able to connect to **`vag-int`** and **`vag-ext`** through Linux Shell.

In this case, the TOE does not perform the authentication and identification of the Maintenance User. This task is responsibility of the operational environment. After that, an access control policy is exercised in order to provide access rights to the Maintenance User.

Application Layer

Application layer of the system is basically responsible for forwarding packets from protocol layer to message layer. For HTTP packets, it consists of built-in malicious data pattern rules that it searches and filters out from the incoming data packets. Clean data packets are registered to system and forwarded to message layer. For SMTP packets, it simply does the registration and forwarding operation.

Message Layer

Message layer of the system is responsible for reorganizing the incoming and outgoing data packets for supported APs.

Crypto/Sign Layer

Crypto sign layer of the system is responsible for the invocation of cryptographic operations functionality in the operating system. With this layer the incoming and outgoing messages are signed/encrypted and verified/decrypted respectively during a regular data flow between internal and external hosts.

Audit

All the activity that takes place in the TOE is audited on disk storage in exclusive locations (on internal disk for **`vag-int`** and on shared disk for **`vag-ext`**) and these audit data is available to be read by administrative via management interface and to be exported by the Maintenance User through the Linux Shell.

Audit functionality of the system is always enabled (i.e., set ON)

Alarm

All the activity that takes place in the OS is recorded by logging components in shared disk in exclusive locations for the internal and external host (out of the scope of TOE). These are collectively referred to as "Logs" of the system.

Alarm module is responsible for checking pre-defined rules over the "Log" and "Audit" data and warns administrative users through management interface in case of a matching condition.

Upon receiving a critical alarm through the system, the TOE will set the system in passive mode.

Disk Access Layer

Main feature of this layer is to read and write to a disk prior to a cryptographic operation.

The modules written in RED font and bold in Figure 2 are in the logical scope of TOE. Modules of the system are detailed as follows:

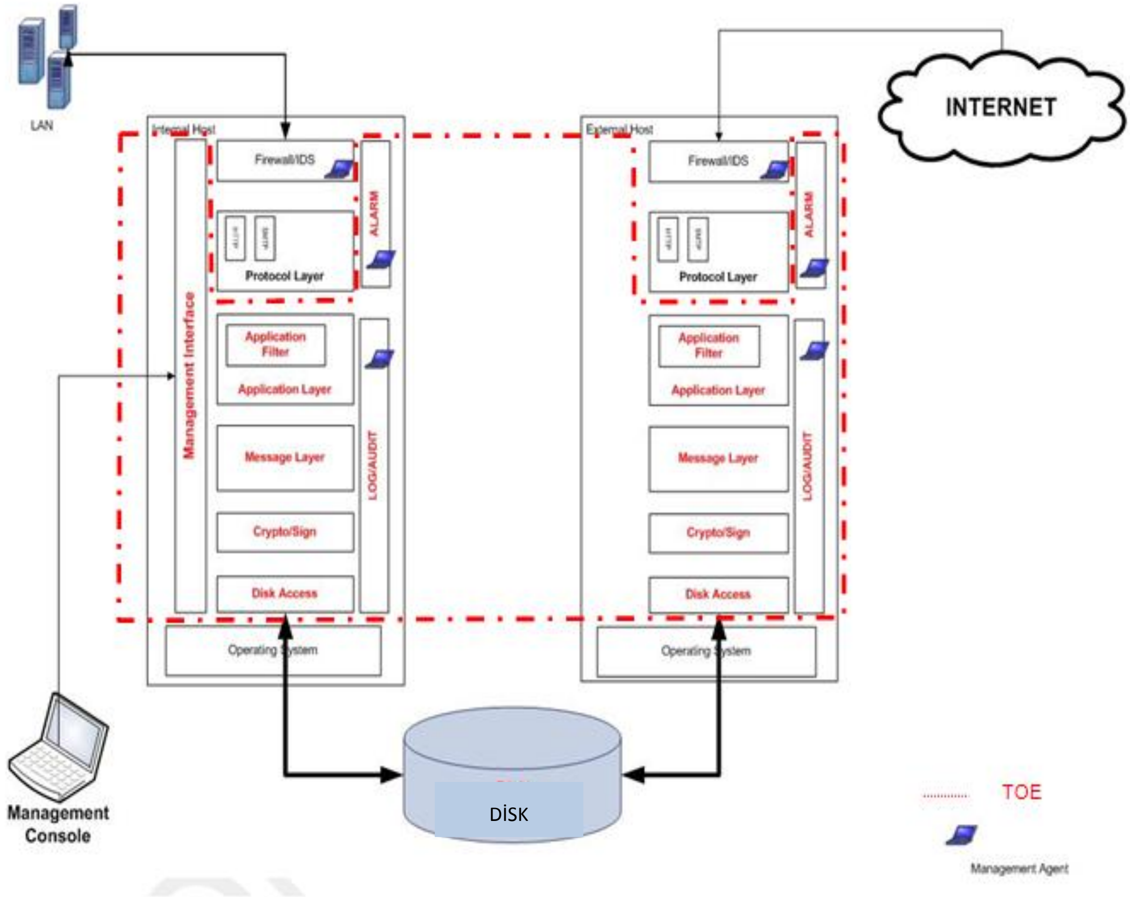


Figure 2 *Logical Scope of TOE*

1.4.2 Physical Scope

Physical software components of the TOE are identified and described in the following table.

#	Module	Description
1	**vag-mgmt-interface**	VAG components running on internal host and client browser, and collectively providing services of management interface to the administrative users.
2	**vag-int, vag-ext**	VAG software including application, message, Crypto/Sign and Disk Access layers (Identical – symmetrical- software having different binary names on internal and external hosts). It also includes the management interface for **vag-int**, and Alarm Log/Audit for both.

#	Module	Description
3	**vag-user-guidance**	VAG documentation (e.g., installation and user manuals) for guidance of administrative users and Maintenance User.

1.4.3 TOE Security Features

The security functionality of the TOE is the following:

- **Audit**: The TOE generates audit logs, and provides the capability of reviewing these audit logs.
- **Alarm**: The TOE includes an automatic procedure to search for predefined attack patterns into the audit logs, and, in case of detecting a potential attack, notify an alarm and act in consequence.
- **Cryptographic operations invocation**: The TOE invokes the operational environment to perform cryptographic operations to cipher and sign the dataflow between **`vag-int`** and **`vag-ext`**.
- **Access control**: The TOE performs an access control for administrative users of the management interface, and an access control for the Maintenance User.
- **Data Importation**: The Maintenance User of the TOE is able to import data into the TOE.
- **Data Exportation**: The Maintenance User of the TOE is able to export data from the TOE.
- **Dataflow Control**: A dataflow access control is performed by the TOE to control the information flow between the external and the internal network.
- **Identification & Authentication**: The TOE performs an Identification and Authentication mechanism for administrative user that access through the management interface.
- **Security Management**: The TOE provides management functionality to users depending on the user role.
- **Security Roles**: The TOE maintains security roles for users.

2. CONFORMANCE CLAIM

This TOE's and ST's conformance claim is stated in the following sub-sections.

2.1 CC Conformance Claim

This TOE and ST are consistent with the following specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 3.1, Revision 3, July 2009, extended.
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements Version 3.1, Revision 3, July 2009, conformant.

2.2 PP and Package Claim

2.2.1 Protection Profile (PP) Claim

This ST makes no conformance claims to any certified Protection Profile.

2.2.2 Package Claim

EAL4+ with the augmentation of ALC_FLR.2 and AVA_VAN.5.

This Security Target elaborated in conformance with "Common Criteria for Information Technology Security Evaluation, Version 3.1 rev 3" contains the IT security requirements of the TOE and specifies the functional and assurance security measures to meet the stated requirements.

2.3 Conformance Rationale

The assurance level of EAL 4+ (ALC_FLR.2, AVA_VAN.5) is considered to be most appropriate for this type of TOE since it is intended to defend against attacks that can be made given the assumptions, organizational security policies and the threats defined in Chapter 3.

3. SECURITY PROBLEM DEFINITION

3.1 Threat Agents

Threats agents of the TOE are described in the table below.

Threat Agents	Description
External VAG User	Any person or software agent sending/receiving IP packets to/from TOE from external network. Attack potential of VAG user is accepted as high.
Internal VAG User	Any person or software agent sending/receiving IP packets to/from TOE from internal network. Attack potential of VAG user is accepted as high.
Authenticated Manager or Operator	An authenticated manager or operator of the management interface. Attack potential of Administrative users is accepted as high.

3.2 Assets

Assets of the TOE are described in the table below. Both confidentiality and integrity of these assets are fundamental for the proper operation of the TOE.

Assets	Description
Administrative User Credentials	Credentials of administrative users for TOE management interface.
Maintenance User Credentials	Credentials of the Maintenance User of the TOE.
Cryptographic Keys	The cryptographic keys that are used for encryption / decryption / digital signature creation and verification.
Audit Data	Audit data stored in the operational environment.
TOE Internal Communication	The communication channel between internal and external hosts of the TOE.
Configuration Data	The configuration data of the TOE and its operational environment.

3.3 Threats

Threats for the TOE are described in the table below.

Threats	Description
T.UNAUTH	An internal VAG user may gain unauthorized access to the TOE through the management interface that causes a loss in the confidentiality and integrity of any of the assets.
T.EAVESDROP	An internal VAG user may follow the traffic between management console and the TOE that cause a loss in the confidentiality of audit data, user credentials and configuration data.
T.OBTAIN	An external VAG user may obtain the TOE's internal communication data being exchanged between external and internal hosts of the TOE, which will cause a loss in the confidentiality of the transmitted data.
T.CRYPTOKEYS	An internal/external VAG user may compromise the cryptographic keys through an unauthorized access to the memory, which will cause a loss in confidentiality, integrity and availability of any of the assets.
T.MEDIATE	An external/internal VAG user may bypass the filtering mechanism of the TOE by compromising the integrity of the configuration data.
T.PRIVILEGES	An authenticated manager or operator of the management interface may gain unauthorized access to resources not allowed for the certain type of administrative user.

3.4 Organizational Security Policies

OSPs	Description
OSP.LOCK	Cryptographic Keys (on USB Flash Disk) must be under the sole control of the Maintenance User.
OSP.AUDIT	The TOE must generate reviewable audit data and all users must be accountable for their actions.

OSPs	Description
OSP.MACP	A **Maintenance Access Control Policy** shall be implemented allowing a specific maintenance role the access to a particular set of maintenance functions. The user holding this maintenanace role is identified and authenticated by the Operating System login terminal (for both **`vag-int`** and **`vag-ext`**). The UserID is to be provided to the TOE to exercise the access control policy. Only the following maintenance functions will be accesible for this role: a. Mount/umount USB Device b. Read license c. Install Patch file d. Modify Password e. Export audit logs f. Export/Restore full backup These maintenance functions will only be accessible for the user holding this role.

3.5 Assumptions

Assumptions	Description
A.PHYSICAL	The TOE is installed in a physically secure location and the only user who can access to the physical location where the TOE is located is the Maintenance User.
A.TIME	The environment provides reliable timestamp.
A.NOEVIL	The administrator of the management interface and the Maintenance User are non-hostile and follow all administrative guidance.
A.SINGEN	The TOE is the only communication channel between internal and external network.
A.PLATFORM	No claims are made on the security of the platform that contains the OS. Compromise of the platform can lead to compromise of TOE.
A.INITIALIZATION	Cryptographic keys must be imported through a secure media during the initialization of the TOE according to a policy.

4. SECURITY OBJECTIVES

4.1 Security Objectives for the TOE

Security Objective	Description
O.AUDIT	The TOE shall generate audit data that can be reviewed by authorized administrative users. All actions performed by a user shall be registered in the audit data.
O.AUTH	The TOE shall authenticate the administrative users before conducting operations through management interface.
O.ALARM	The TOE shall inspect audit / log data in order to generate alarms, and act in consequence.
O.ACCESSCONTROL	The TOE shall implement a **Management interface access control policy** to provide authorization to the administrative users according to their role.
O.FLOW	The TOE shall control the information flow between internal and external network.
O.CRYPTOOP	The TOE shall invoke the operating environment to encrypt and sign the communication between its internal and external hosts.
O.MACP	The TOE shall implement a **Maintenance Access Control Policy** to restrict a specific maintenance role, the access to the following maintenance functions (for both **`vag-int`** and **`vag-ext`**): a. Mount/umount USB Device b. Read license c. Install Patch file d. Modify Password e. Export audit logs f. Export/Restore full backup These maintenance functions shall only be accessible for the user holding this role. The implementation of the access control policy will be based on the UserID attribute provided by the operating system login mechanism.

4.2 Security Objectives for the Operational Environment

Security Objective	Description
OE.PHYSECURE	The TOE must be kept in a physically secured location to prevent attacker from physically accessing the TOE.
OE.NOEVIL	The administrator of the management interface and the Maintenance User are non-hostile, appropriately trained, and follow all user guidance, installation guidance and configuration guidance.
OE.TIME	The operational environment shall provide a reliable date and timestamp from trusted source.
OE.SINGEN	Owners of the TOE must ensure that TOE is the only connection between the internal and external network.
OE.PLATFORM	The platform that runs TOE shall be protected against compromise.
OE. INITIALIZATION	Maintenance User of the TOE must ensure that importing the cryptographic keys via a secure media will initialize TOE, and this secure media will be under the sole control of this user.
OE.SECURECOMMUNICATION	The Operational Environment shall provide a secure communication line between the TOE and the Management Console.
OE.CRYPTOOP	The Operational Environment shall provide encryption and signature services to the TOE.
OE.USERID_PROVIDER	The operating system login mechanism in both **`vag-int`** and **`vag-ext`** shall identify and authenticate the user and provide the UserID to the TOE for the purpose of exercising the **Maintenance Access Control Policy** referred in the OSP.MACP organisational policy.

4.3 Security Objectives Rationale

The following table is showing the mappings between security objectives and threats/assumptions and security policies. The table is also stating the rationales for the mappings.

Threat / Policy / Assumption	Security Objective	Rationale
T.UNAUTH	O.AUTH	The objective of O.AUTH guarantees that only authenticated administrative user can access to the management interface via management console.
T.EAVESDROP	OE.SECURECOMMUNICATION	The objective of OE.SECURECOMMUNICATION is to prevent eavesdropping and similar type of internal attacks during the communication between management console and management interface of the TOE.
T.OBTAIN	O.CRYPTOOP OE.CRYPTOOP	The objective of O.CRYPTOOP is to invoke the cryptographic operation functions provided by the operating environment. The objective of OE.CRYPTOOP is to ensure confidentiality and integrity of user data during the transfer between internal and the external host of the TOE by encryption and signature.
T.CRYPTOKEYS	O.ACCESSCONTROL O.FLOW OE.PHYSECURE OE.NOEVIL	The objective of O.ACCESSCONTROL is to ensure that the access control policy for administrative users through the management interface will be exercised to avoid access to the cryptokeys. The objective O.FLOW controls the usage of the cryptokeys. OE.PHYSECURE guarantees that no physical access to the location where VAG is deployed is available for unauthorized personnel, and therefore no unauthorized access to the VAG memory is possible, and OE.NOEVIL indicates that the unique users that can access to the physical location where VAG is deployed is trusted.

Threat / Policy / Assumption	Security Objective	Rationale
T.MEDIATE	O.FLOW O.ALARM O.AUDIT	The objective of O.FLOW is to control the information flow between internal and external network and allow only approved data flow. The objective of O.ALARM is to assure a mechanism for automatic audit review in order to prevent predefined attack patterns, and act in case of attack detection. The objective of O.AUDIT is to assure that necessary audit logs will be generated and recorded for satisfaction of O.ALARM.
T.PRIVILEGES	O.ACCESSCONTROL	The objective of O.ACCESSCONTROL guarantees that authorized administrative users can read/modify data through management interface according to their access rights.
OSP.AUDIT	O.AUDIT	The objective of O.AUDIT is to associate each event with a user where applicable.
OSP.MACP	O.MACP OE.USERID_PROVIDER	The OSP.MACP requires the implementation of an access control policy to restrict the accessibility to a set of maintenance functions directly through the **vag-int** or **vag-ext**. The TOE implements the access control policy for the set of functions as defined in O.MACP using the UserID of the user authenticated by the operating system. This UserID is to be provided to the TOE by the operating system (OE.USERID_PROVIDER).
A.PHYSICAL	OE.PHYSECURE	This objective is to ensure that TOE will be protected against physical attacks.

Threat / Policy / Assumption	Security Objective	Rationale
A.TIME	OE.TIME	This objective assures operational environment to provide reliable timestamps.
A.NOEVIL	OE.NOEVIL	This objective ensures that the administrator of the management interface and the Maintenance User are trained and do not intentionally cause threats on TOE.
A.SINGEN	OE.SINGEN	This objective ensures that TOE cannot be bypassed during communication with external network.
A.INITIALIZATION OSP.LOCK	OE.INITIALIZATION	This objective ensures that Maintenance User of TOE initializes TOE with the correct cryptokeys according to user guidance, and that the secure media containing the cryptokeys will be under the sole control of this user.
A.PLATFORM	OE.PLATFORM	This objective states that TOE is not protected against a compromise in the operational environment.

5. EXTENDED COMPONENTS DEFINITION

5.1 Class FCS: Cryptographic Support

5.1.1 Cryptographic Operation

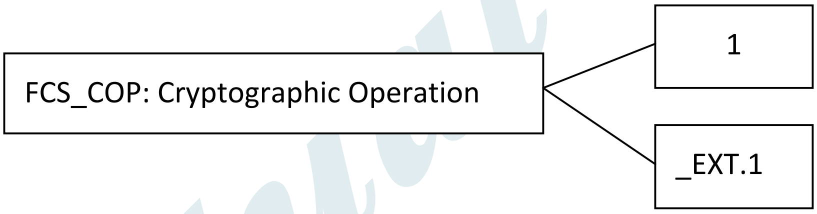
Family Behavior

In order for a cryptographic operation to function correctly, the operation must be performed in accordance with a specified algorithm and with a cryptographic key of a specified size. This family should be included whenever there are requirements for cryptographic operations to be performed.

Typical cryptographic operations include data encryption and/or decryption, digital signature generation and/or verification, cryptographic checksum generation for integrity and/or verification of checksum, secure hash (message digest), cryptographic key encryption and/or decryption, and cryptographic key agreement.

Component Leveling

The components of this family is extended with one component FCS_COP_EXT.1



FCS_COP.1 Cryptographic operation requires a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes. The specified algorithm and cryptographic key sizes can be based on an assigned standard.

FCS_COP_EXT.1 Cryptographic operation requires a cryptographic operation to be performed by operational environment by invocation of the TOE with a specified algorithm and with a cryptographic key of specified sized. The specified algorithm and cryptographic key sized can be based on an assigned Standard.

Management FCS_COP_EXT.1:

There are no management activities foreseen.

Audit FCS_COP_EXT.1:

a) Minimal: Success and failure, and the type of cryptographic operation.

b) Basic: Any applicable cryptographic mode(s) of operation, subject attributes and object attributes.

FCS_COP_EXT.1 Cryptographic operation invocation

Hierarchical to: No other components.

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

FCS_COP_EXT.1.1 The TOE shall invoke the Operational Environment to perform encrypt/decrypt with a specified symmetric cryptographic algorithm AES and cryptographic key sizes 256 as provided by OpenSSL.

FCS_COP_EXT.1.2 The TOE shall invoke the Operational Environment to perform sign/verify with a specified asymmetric cryptographic algorithm DSA and cryptographic key sizes 1024 as provided by OpenSSL.

> ***Application Note - 1:*** *The family of Cryptographic Operation under the Cryptographic Support class is extended with FCS_COP_EXT.1 because the cryptographic operation performed by TOE is only consisting of the invocation. The operational environment conducts all the requirements of FCS_COP.1, however the description of the requirement is enforcing TOE to perform the operation. With this extended component under the same family, TOE performs the invocation and the operational environment conducts the cryptographic operation.*

> ***Application Note - 2:*** *The dependency for FCS_CKM.4 is not met for this extended component since the TOE does not destruct the keys. When new cryptographic keys are introduced, they will be imported according to the policy and the updated keys will be obfuscated in the memory for secure storage.*

6. SECURITY REQUIREMENTS

6.1 Security Functional Requirements

The following table summarizes the Security Functional Requirements included in this Security Target including their dependencies. The first column indicates the certain SFR, the second one specifies the dependencies for the certain SFR, and the third column indicates the way the dependencies have been satisfied. For the non-satisfied dependencies, a rationale (identified by a cardinal) is included in the last row of the table.

SFR	Dependencies	Satisfied
FAU_ARP.1	FAU_SAA.1	Y (FAU_SAA.1)
FAU_GEN.1	FPT_STM.1	N **(1)**
FAU_GEN.2	FAU_GEN.1 FIA_UID.1	Y (FAU_GEN.1) Y (FIA_UID.2) N **(2)**
FAU_SAA.1	FAU_GEN.1	Y (FAU_GEN.1)
FAU_SAR.1	FAU_GEN.1	Y (FAU_GEN.1)
FAU_SAR.2	FAU_SAR.1	Y (FAU_SAR.1)
FCS_COP_EXT.1	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	N **(3)** N **(4)**
FDP_ACC.1/MANAGEMENT	FDP_ACF.1	Y (FDP_ACF.1/MANAGEMENT)
FDP_ACF.1/MANAGEMENT	FDP_ACC.1 FMT_MSA.3	Y (FDP_ACC.1/MANAGEMENT) Y (FMT_MSA.3)
FDP_ACC.1/MAINTENANCE	FDP_ACF.1	Y (FDP_ACF.1/MAINTENANCE)
FDP_ACF.1/MAINTENANCE	FDP_ACC.1 FMT_MSA.3	Y (FDP_ACC.1/MAINTENANCE) N **(5)**
FDP_ETC.1	[FDP_ACC.1 or FDP_IFC.1]	Y (FDP_ACC.1/MAINTENANCE)
FDP_IFC.1	FDP_IFF.1	Y (FDP_IFF.1)
FDP_IFF.1	FDP_IFC.1 FMT_MSA.3	Y (FDP_IFC.1) N **(6)**
FDP_ITC.1	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	Y (FDP_ACC.1/MAINTENANCE) N **(5.1)**
FIA_AFL.1	FIA_UAU.1	Y (FIA_UAU.2)
FIA_ATD.1	None.	N/A
FIA_SOS.1	None.	N/A
FIA_UAU.2	FIA_UID.1	Y (FIA_UID.2) N **(2)**
FIA_UID.2	None.	N/A

SFR	Dependencies	Satisfied
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1 FMT_SMF.1	Y (FDP_ACC.1/MANAGEMENT) Y (FMT_SMR.1) Y (FMT_SMF.1)
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Y (FMT_MSA.1) Y (FMT_SMR.1)
FMT_SMF.1	None.	N/A
FMT_SMR.1	FIA_UID.1	Y (FIA_UID.2) N **(5)**

Non-Satisfied Dependencies Rationale

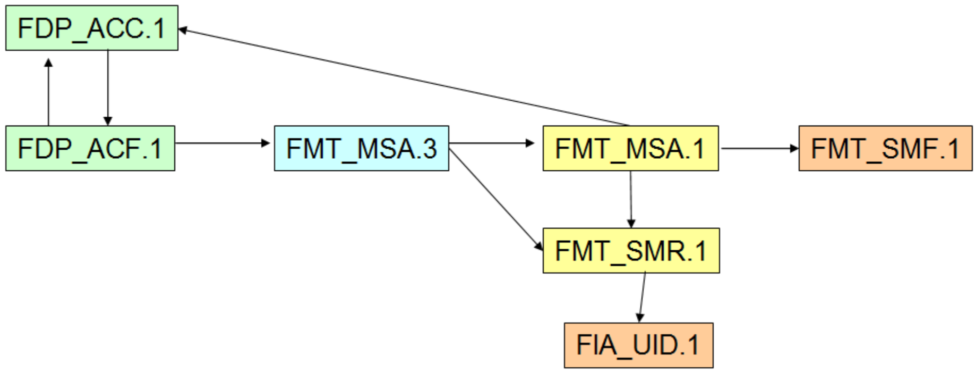
(1) The dependency for FPT_STM.1 Reliable Time Stamps is not met given that the TOE does not generate time stamps for audit logs. As stated in OE.TIME, the responsible for the generation of time stamps is the operational environment.

(2) The user identification for the Maintenance User is not performed by the TOE. As stated in OE.USERID_PROVIDER, the operational environment is the responsible of providing the user ID of the Maintenance User to the TOE.

(3) The importation of the keys used for the cryptographic operations belongs to the initialization and start-up of the system and is defined as part of the TOE security architecture.

(4) The dependency for FCS_CKM.4 is not met for this extended component since the keys are not destructed by the TOE, given that the keys are constantly used to perform the cryptographic operations defined in FCS_COP_EXT.1.

(5) The FDP_ACC.1 and FDP_ACF.1 used to define an access control policy require the existence by dependencies of the following SFRs as depicted in the following chart:



The following dependencies have not been satisfied in the frame of the specification of the **maintenance access control policy**:

(5.1) FMT_MSA.3 & FMT_MSA.1 (FDP_ACC.1 and FDP_ACF.1). The policy attribute -UserID of the subject- is provided by the OS and not managed by the TOE.

(5.2) FIA_UID.1 (FMT_SMR.1). Although FMT_MS3.1 is not satisfied, the TOE maintains a set of roles including the ***maintenance*** role involved in this policy. For the user holding this role and allowed to access the objects defined in this policy, the UserID attribute needed is provided by the operating system (out of the scope of the evaluation) obtained in its login process (see OE.USERID_PROVIDER). Therefore the dependency with FIA_UID.1 is not satisfied.

(6) The dependency with FMT_MSA.3 is not satisfied, given that the security attributes of the **data flow control policy** are not configurable, and therefore, no management operations over security attributes for this data flow control policy is be performed.

6.1.1 Security Audit

FAU_ARP.1 Security alarms

FAU_ARP.1.1 The TSF shall take [**generate log, send notification for all alarms and stop data flow (put the system in passive mode) for critical alarms**] (i) upon detection of a potential security violation and (ii) when the disk size reaches the configured percentage of its total capacity.

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 [**minimum**] level of audit; and

c) [**the following Audited Events**

- **System alarms**
- **Actions derived from potential security violations**
- **Proxy accepted requests**
- **Proxy denied requests**
- **Users access to the TOE**
- **Unsuccessful authentication attempts**
- **Dataflow filters results**
- **All management functionality of FMT_SMF.1**

]

Application Note: *Start-up of audit functions is done at the TOE initialization phase, and then is never shutdown while the TOE is up and running; i.e., audit functionality is always ON.*

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **[none]**.

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.

Application Note: *The TOE associates users with the auditable events if the event is conducted by an administrative user or the Maintenance User. Otherwise the TOE will only log the event itself, such as the information flow between internal and external networks.*

FAU_SAA.1 Potential violation analysis

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

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

a) Accumulation or combination of **[**

Rule Name	***Definition***
ALRM_CRIT	**Any message by any TOE component containing "CRIT" or "VAG_CRIT" keyword** **Any message by any TOE (Management) component containing "MGMT_CRIT" and "Critical Disk Space"keyword**
ALRM_ERR	**Any message by any TOE component containing "ERROR" or "VAG_ERR" keyword**
ALRM_LOG	**Any message by any component containing "VAG_ALRT" or "VAG_CRIT" or "VAG_ERR" or "VAG_BUG" or "ERROR" or "CRIT" keyword**
ALRM_VAG	**Any message by VAG containing "[VAG]" keyword**
ALRM_MGMT	**Any message by any TOE (Management) component**
ALRM_LOGIN	**Any message by OS containing "login" keyword**
ALRM_HIDS	**Any message by HIDS software containing keyword "CRIT" or "ERROR"**
ALRM_NIDS	**Any message by NIDS software**
ALRM_KERNEL	**Any message by OS containing "kernel:" keyword**

] known to indicate a potential security violation;

b) **[none]**.

FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide [**all administrative users**] with the capability to read [**list of audited events in FAU_GEN.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.

FAU_SAR.2 Restricted audit review

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

6.1.2 Cryptographic Support

FCS_COP_EXT.1 Cryptographic Operation Invocation

FCS_COP_EXT.1.1 The TOE shall invoke the Operational Environment to perform encrypt/decrypt with a specified symmetric cryptographic algorithm AES and cryptographic key sizes 256 as provided by OpenSSL.

FCS_COP_EXT.1.2 The TOE shall invoke the Operational Environment to perform sign/verify with a specified asymmetric cryptographic algorithm DSA and cryptographic key sizes 1024 as provided by OpenSSL.

Application Note - 1: *The family of Cryptographic Operation under the Cryptographic Support class is extended with FCS_COP_EXT.1 because the cryptographic operation performed by TOE is only consisting of the invocation. The operational environment conducts all the requirements of FCS_COP.1, however the description of the requirement is enforcing TOE to perform the operation. With this extended component under the same family, TOE performs the invocation and the operational environment conducts the cryptographic operation.*

Application Note - 2: *The dependency for FCS_CKM.4 is not met for this extended component since the keys are not destructed by the TOE. When new cryptographic keys are introduced, they will be imported according to the policy and the updated keys will be obfuscated in the memory for secure storage.*

6.1.3 User Data Protection

FDP_ACC.1/MANAGEMENT Subset access control

FDP_ACC.1.1 The TSF shall enforce the [**management interface access control policy**] on [

List of Subjects;

Administrative Users

List of Objects;

System Information

Configuration Data

Other Administrative Users

Administrative User List

Own Password

Audit Logs

Partial Backups

List of Operations;

Read

Modify

Delete

Create

Restore

].

FDP_ACF.1/MANAGEMENT Security based access control

FDP_ACF.1.1 The TSF shall enforce the [**management interface access control policy**] to objects based on the following: [

List of Subjects;

Administrative Users

List of Objects;

System Information

Configuration Data

Other Administrative Users

Administrative User List

Own Password

Audit Logs

Partial Backups

List of Security Attributes for Subjects:

Administrative user Role

List of Security Attributes for Objects:

None

].

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

The list of subjects is only granted access for list of operations on the list of objects according to their security attributes shown below;

	System Information	Configuration Data	Other Administrative Users	Administrative user list	Own Password	Audit Logs	Partial Backups
Administrator	Read	Read Modify	Create Modify Delete	Read	Modify	Read	Read Create Restore
Manager	Read	Read Modify	NA	Read	Modify	Read	Read Create
Operator	Read	Read[*]	NA	NA	Modify	Read	NA

].

Application Note: *The privileges are to be interpreted as follows:*

Read: Can read the content.

Modify: Can modify the content.

Create: Can create an instance of the object (a user or a partial backup).

Delete: Can delete an instance of an object.

Restore: Can restore a backup.

NA: Subject has no privilege over the object.

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [**none**].

Application Note (*): *The operator user can only read (i) the IDS status, (ii) the web status and parameters, (iii) the mail status and parameters from Configuration data.*

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**none**].

Application Note: *The modification of "Other Administrative Users" includes the password change and the role change.*

[*] *See Application Note in FDP_ACF.1.3*

FDP_ACC.1/MAINTENANCE Subset access control

FDP_ACC.1.1 The TSF shall enforce the [**maintenance access control policy**] on [

List of Subjects;

Maintenance User

List of Objects;

USB Device

License

Patch files

Own Password

Audit Logs

Full Backups

List of Operations;

Read

Mount

Unmount

Modify

Install

Export

Restore

].

FDP_ACF.1/ MAINTENANCE Security based access control

FDP_ACF.1.1 The TSF shall enforce the [**maintenance access control policy**] to objects based on the following: [

List of Subjects;

Maintenance User

List of Objects;

USB Device

License

Patch files

Own Password

Audit Logs

Full Backups

List of Security Attributes for Subjects:

User ID

List of Security Attributes for Objects:

None

].

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

The list of subjects is only granted access for list of operations on the list of objects according to their security attributes shown below;

	USB Device	License	Patch Files	Own Password	Audit Logs	Full Backup
Maintenance User	Mount Unmount	Read	Install	Modify	Export	Export Restore

].

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [**none**].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**none**].

Application Note - 1: *As detailed in the SPD (OSP.MACP), the entity responsible for the Maintenance User Identification and Authentication is the Debian/Linux Operating System, which does not belong to the TOE. Once the Identification and Authentication has been successfully carried out by the Debian/Linux Operating System, the User ID is provided to the TOE that uses it to exercise the access control policy.*

Application Note - 2: *The Maintenance User can access both parts of the TOE,* **`vag-int`** *and* **`vag-ext`***, and therefore, this access control policy is applicable to both interfaces.*

FDP_ETC.1 Export of user data without security attributes

FDP_ETC.1.1 The TSF shall enforce the [**maintenance access control policy**] when exporting user data, controlled under the SFP(s), outside of the TOE.

FDP_ETC.1.2 The TSF shall export the user data without the user data's associated security attributes.

Application Note: *The TOE provides user data protection while exporting audit logs and full backups. This action can be performed by the Maintenance User through the Linux Shell.*

FDP_IFC.1 Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the [**data flow control policy**] on [

List of Subjects:

VAG Users

Information:

All data flow between internal and external network through the TOE

Operation:

Data Input/Output

].

FDP_IFF.1 Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the [**data flow control policy**] based on the following types of subject and information security attributes: [

List of Subjects:

VAG Users

List of Subject Attributes:

Source IP Address

List of Information:

All data flow between internal and external network through the TOE

List of Information Attributes:

Destination IP Address

HTTP Header Contents

SMTP Header Contents

].

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: [

For the data input, the access is granted if the Source IP address is allowed to access to the Destination IP address, and if the HTTP or SMTP header pass through the application filter.

For the data output, the access is granted if the Source IP address is allowed to access to the Destination IP address, and if the HTTP or SMTP header pass through the application filter.

].

FDP_IFF.1.3 The TSF shall enforce the [**none**].

FDP_IFF.1.4 The TSF shall explicitly authorize an information flow based on the following rules: [**none**].

FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [**none**].

FDP_ITC.1 Import of user data without security attributes

FDP_ITC.1.1 The TSF shall enforce the [**maintenance access control policy**] when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [**none**].

Application Note: *The TOE provides user data protection while importing patch files and full backups. This action can be performed by the Maintenance User through the Linux Shell.*

6.1.4 Identification and Authentication

FIA_AFL.1 Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when [**3**] unsuccessful authentication attempts occur related to [**authentication for administrative users (administrator, operator, manager)**].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [**met**], the TSF shall [**lock the administrative user account**].

Application Note - 1: *This requirement is only applicable for administrative users that access the TOE through the management interface.*

Application Note - 2: *The Administrator have the capability of unlocking Managers or Operators.*

Application Note - 3: *Unlocking the Administrator is only possible by a reboot of the system. However, once the system is rebooted, all administrative users (including the Administrator) are unlocked.*

FIA_ATD.1 User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [**administrative user role, administrative user password**].

> ***Application Note:*** *This requirement is only applicable for administrative users that access the TOE through the management interface.*

FIA_SOS.1 Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [**the following metric: the minimal password length is 8 characters**].

> ***Application Note:*** *This requirement is only applicable for administrative users that access the TOE through the management interface.*

FIA_UAU.2 User authentication before any action

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

> ***Application Note:*** *The authentication of administrative users is performed by the TOE. The authentication of the Maintenance User is performed by the environment (Debian/Linux).*

FIA_UID.2 User identification before any action

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

> ***Application Note:*** *The identification of administrative users is performed by the TOE. The identification of the Maintenance User is performed by the environment (Debian/Linux).*

6.1.5 Security Management

FMT_MSA.1 Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [**management interface access control policy**] to restrict the ability to [**<u>change default</u>**, **<u>modify</u>**] the security attributes [**administrative user role**] to [**administrator**].

FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the [**management interface access control policy**] to provide [**<u>permissive</u>**] default values for security attributes that are used to enforce the SFP.

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

FMT_SMF.1 Specification of management functions

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

- **Configuration Data management**
- **Administrative Users management**
- **Passwords management**
- **Full Backups management**
- **Partial Backups management**
- **Audit logs management**
- **Patch management**

- **USB device management**
- **Read license**

].

FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles [**administrator, manager, operator, maintenance**].

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

> ***Application Note - 1:*** *The unique kind of users that can be created and associated to a role during the normal operation of the TOE is the administrative users. The administrator is able to associate the operator or manager role to new administrative users.*
>
> ***Application Note - 2:*** *The maintenance role contemplates only a user, the Maintenance User, and there is no way of creating more users associated to this role.*
>
> ***Application Note - 3:*** *There is a virtual user called "LOGWATCH" observed only in management interface alarm messages area, which is not an actual user but just an internal process identifier.*

6.2 Security Assurance Requirements

The Security Assurance Requirements for the TOE are the Evaluation Assurance Level 4 augmented with ALC_FLR.2 and AVA_VAN.5. EAL4 is the highest mutually recognized level and TOE could be able to comply. The requirements for this level are listed below;

Assurance Class	Assurance Component
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_FLR.2 – Flaw reporting procedures
	ALC_LCD.1 – Development defined life-cycle model
	ALC_TAT.1 – Well-defined development tools
ASE:	ASE_CCL.1 – Conformance claims

Assurance Class	Assurance Component
Security Target Evaluation	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.5 – Advanced methodical vulnerability analysis

Table 1 Security Assurance Requirements

6.2.1 ADV_ARC.1 Security Architecture Description

Dependencies satisfied by: ADV_FSP.4 Complete functional specification
ADV_TDS.3 Basic modular design

Developer action elements:

6.2.1.1 ADV_ARC.1.1D

The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

6.2.1.2 ADV_ARC.1.2D

The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

6.2.1.3 ADV_ARC.1.3D

The developer shall provide a security architecture description of the TSF.

Content and presentation of evidence elements:

6.2.1.4 ADV_ARC.1.1C

The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

6.2.1.5 ADV_ARC.1.2C

The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

6.2.1.6 ADV_ARC.1.3C

The security architecture description shall describe how the TSF initialization process is secure.

6.2.1.7 ADV_ARC.1.4C

The security architecture description shall demonstrate that the TSF protects itself from tampering.

6.2.1.8 ADV_ARC.1.5C

The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

6.2.2 ADV_FSP.4 Complete functional specification

Dependencies satisfied by: ADV_TDS.3 Basic modular design

Developer action elements:

6.2.2.1 ADV_FSP.4.1D

The developer shall provide a functional specification.

6.2.2.2 ADV_FSP.4.2D

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

Content and presentation of evidence elements:

6.2.2.3 ADV_FSP.4.1C

The functional specification shall completely represent the TSF.

6.2.2.4 ADV_FSP.4.2C

The functional specification shall describe the purpose and method of use for all TSFI.

6.2.2.5 ADV_FSP.4.3C

The functional specification shall identify and describe all parameters associated with each TSFI.

6.2.2.6 ADV_FSP.4.4C

The functional specification shall describe all actions associated with each TSFI.

6.2.2.7 ADV_FSP.4.5C

The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

6.2.2.8 ADV_FSP.4.6C

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

6.2.3 ADV_IMP.1 Implementation representation of the TSF

Dependencies satisfied by: ADV_TDS.3 Basic modular design
ALC_TAT.1 Well-defined development tools

Developer action elements:

6.2.3.1 ADV_IMP.1.1D

The developer shall make available the implementation representation for the entire TSF.

6.2.3.2 ADV_IMP.1.2D

The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.

Content and presentation of evidence elements:

6.2.3.3 ADV_IMP.1.1C

The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.

6.2.3.4 ADV_IMP.1.2C

The implementation representation shall be in the form used by the development personnel.

6.2.3.5 ADV_IMP.1.3C

The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

6.2.4 ADV_TDS.3 Basic modular design

Dependencies satisfied by: ADV_FSP.4 Complete functional specification

Developer action elements:

6.2.4.1 ADV_TDS.3.1D

The developer shall provide the design of the TOE.

6.2.4.2 ADV_TDS.3.2D

The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

Content and presentation of evidence elements:

6.2.4.3 ADV_TDS.3.1C

The design shall describe the structure of the TOE in terms of subsystems.

6.2.4.4 ADV_TDS.3.2C

The design shall describe the TSF in terms of modules.

6.2.4.5 ADV_TDS.3.3C

The design shall identify all subsystems of the TSF.

6.2.4.6 ADV_TDS.3.4C

The design shall provide a description of each subsystem of the TSF.

6.2.4.7 ADV_TDS.3.5C

The design shall provide a description of the interactions among all subsystems of the TSF.

6.2.4.8 ADV_TDS.3.6C

The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

6.2.4.9 ADV_TDS.3.7C

The design shall describe each SFR-enforcing module in terms of its purpose and relationship with other modules.

6.2.4.10 ADV_TDS.3.8C

The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, interaction with other modules and called SFR-related interfaces to other SFR-enforcing modules.

6.2.4.11 ADV_TDS.3.9C

The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.

6.2.4.12 ADV_TDS.3.10C

The mapping shall demonstrate that all TSFIs trace to the behavior described in the TOE design that they invoke.

6.2.5 AGD_OPE.1 Operational user guidance

Dependencies satisfied by: ADV_FSP.4 Complete functional specification

Developer action elements:

6.2.5.1 AGD_OPE.1.1D

The developer shall provide operational user guidance.

Content and presentation of evidence elements:

6.2.5.2 AGD_OPE.1.1C

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

6.2.5.3 AGD_OPE.1.2C

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

6.2.5.4 AGD_OPE.1.3C

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

6.2.5.5 AGD_OPE.1.4C

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

6.2.5.6 AGD_OPE.1.5C

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

6.2.5.7 AGD_OPE.1.6C

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

6.2.5.8 AGD_OPE.1.7C

The operational user guidance shall be clear and reasonable.

6.2.6 AGD_PRE.1 Preparative procedures

Dependencies satisfied by: No dependencies.

Developer action elements:

6.2.6.1 AGD_PRE.1.1D

The developer shall provide the TOE including its preparative procedures.

Content and presentation of evidence elements:

6.2.6.2 AGD_PRE.1.1C

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

6.2.6.3 AGD_PRE.1.2C

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

6.2.7 ALC_CMC.4 Production support, acceptance procedures and automation

Dependencies satisfied by: ALC_CMS.4 Problem tracking CM coverage

ALC_DVS.1 Identification of security measures

ALC_LCD.1 Developer defined life-cycle model

Developer action elements:

6.2.7.1 ALC_CMC.4.1D

The developer shall provide the TOE and a reference for the TOE.

6.2.7.2 ALC_CMC.4.2D

The developer shall provide the CM documentation.

6.2.7.3 ALC_CMC.4.3D

The developer shall use a CM system.

Content and presentation of evidence elements:

6.2.7.4 ALC_CMC.4.1C

The TOE shall be labeled with its unique reference.

6.2.7.5 ALC_CMC.4.2C

The CM documentation shall describe the method used to uniquely identify the configuration items.

6.2.7.6 ALC_CMC.4.3C

The CM system shall uniquely identify all configuration items.

6.2.7.7 ALC_CMC.4.4C

The CM system shall provide automated measures such that only authorized changes are made to the configuration items.

6.2.7.8 ALC_CMC.4.5C

The CM system shall support the production of the TOE by automated means.

6.2.7.9 ALC_CMC.4.6C

The CM documentation shall include a CM plan.

6.2.7.10 ALC_CMC.4.7C

The CM plan shall describe how the CM system is used for the development of the TOE.

6.2.7.11 ALC_CMC.4.8C

The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

6.2.7.12 ALC_CMC.4.9C

The evidence shall demonstrate that all configuration items are being maintained under the CM system.

6.2.7.13 ALC_CMC.4.10C

The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

6.2.8 ALC_CMS.4 Problem tracking CM coverage

Dependencies satisfied by: No dependencies.

Developer action elements:

6.2.8.1 ALC_CMS.4.1D

The developer shall provide a configuration list for the TOE.

Content and presentation of evidence elements:

6.2.8.2 ALC_CMS.4.1C

The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.

6.2.8.3 ALC_CMS.4.2C

The configuration list shall uniquely identify the configuration items.

6.2.8.4 ALC_CMS.4.3C

For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

6.2.9 ALC_DEL.1 Delivery procedures

Dependencies satisfied by: No dependencies.

Developer action elements:

6.2.9.1 ALC_DEL.1.1D

The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

6.2.9.2 ALC_DEL.1.2D

The developer shall use the delivery procedures.

Content and presentation of evidence elements:

6.2.9.3 ALC_DEL.1.1C

The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

6.2.10 ALC_DVS.1 Identification of security measures

Dependencies satisfied by: No dependencies.

Developer action elements:

6.2.10.1 ALC_DVS.1.1D

The developer shall produce and provide development security documentation.

Content and presentation of evidence elements:

6.2.10.2 ALC_DVS.1.1C

The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

6.2.11 ALC_FLR.2 Flaw reporting procedures

Dependencies satisfied by: No dependencies

Developer action elements:

6.2.11.1 ALC_FLR.2.1D

The developer shall document and provide flaw remediation procedures addressed to TOE developers.

6.2.11.2 ALC_FLR.2.2D

The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

6.2.11.3 ALC_FLR.2.3D

The developer shall provide flaw remediation guidance addressed to TOE users.

Content and presentation of evidence elements:

6.2.11.4 ALC_FLR.2.1C

The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

6.2.11.5 ALC_FLR.2.2C

The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

6.2.11.6 ALC_FLR.2.3C

The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

6.2.11.7 ALC_FLR.2.4C

The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

6.2.11.8 ALC_FLR.2.5C

The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

6.2.11.9 ALC_FLR.2.6C

The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

6.2.11.10 ALC_FLR.2.7C

The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

6.2.11.11 ALC_FLR.2.8C

The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

6.2.12 ALC_LCD.1 Developer defined life-cycle model

Dependencies satisfied by: No dependencies.

Developer action elements:

6.2.12.1 ALC_LCD.1.1D

The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

6.2.12.2 ALC_LCD.1.2D

The developer shall provide life-cycle definition documentation.

Content and presentation of evidence elements:

6.2.12.3 ALC_LCD.1.1C

The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

6.2.12.4 ALC_LCD.1.2C

The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

6.2.13 ALC_TAT.1 Well-defined development tools

Dependencies satisfied by: ADV_IMP.1 Implementation representation of the TSF

Developer action elements:

6.2.13.1 ALC_TAT.1.1D

The developer shall provide the documentation identifying each development tool being used for the TOE.

6.2.13.2 ALC_TAT.1.2D

The developer shall document and provide the selected implementation-dependent options of each development tool.

Content and presentation of evidence elements:

6.2.13.3 ALC_TAT.1.1C

Each development tool used for implementation shall be well-defined.

6.2.13.4 ALC_TAT.1.2C

The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

6.2.13.5 ALC_TAT.1.3C

The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

6.2.14 ASE_INT.1 ST Introduction

Dependencies satisfied by: No dependencies.

Developer action elements:

6.2.14.1 ASE_INT.1.1D

The developer shall provide an ST introduction.

Content and presentation of evidence elements:

6.2.14.2 ASE_INT.1.1C

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

6.2.14.3 ASE_INT.1.2C

The ST reference shall uniquely identify the ST.

6.2.14.4 ASE_INT.1.3C

The TOE reference shall identify the TOE.

6.2.14.5 ASE_INT.1.4C

The TOE overview shall summarize the usage and major security features of the TOE.

6.2.14.6 ASE_INT.1.5C

The TOE overview shall identify the TOE type.

6.2.14.7 ASE_INT.1.6C

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

6.2.14.8 ASE_INT.1.7C

The TOE description shall describe the physical scope of the TOE.

6.2.14.9 ASE_INT.1.8C

The TOE description shall describe the logical scope of the TOE.

6.2.15 ASE_CCL.1 Conformance claims

Dependencies satisfied by: ASE_INT.1 ST Introduction

ASE_ECD.1 Extended components definition

ASE_REQ.2 Derived security requirements

Developer action elements:

6.2.15.1 ASE_CCL.1.1D

The developer shall provide a conformance claim.

6.2.15.2 ASE_CCL.1.2D

The developer shall provide a conformance claim rationale.

Content and presentation of evidence elements:

6.2.15.3 ASE_CCL.1.1C

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

6.2.15.4 ASE_CCL.1.2C

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

6.2.15.5 ASE_CCL.1.3C

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

6.2.15.6 ASE_CCL.1.4C

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

6.2.15.7 ASE_CCL.1.5C

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

6.2.15.8 ASE_CCL.1.6C

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

6.2.15.9 ASE_CCL.1.7C

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

6.2.15.10 ASE_CCL.1.8C

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

6.2.15.11 ASE_CCL.1.9C

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

6.2.15.12 ASE_CCL.1.10C

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

6.2.16 ASE_SPD.1 Security problem definition

Dependencies satisfied by: No dependencies.

Developer action elements:

6.2.16.1 ASE_APD.1.1D

The developer shall provide a security problem definition.

Content and presentation of evidence elements:

6.2.16.2 ASE_SPD.1.1C

The security problem definition shall describe the threats.

6.2.16.3 ASE_SPD.1.2C

All threats shall be described in terms of a threat agent, an asset, and an adverse action.

6.2.16.4 ASE_SPD.1.3C

The security problem definition shall describe the OSPs.

6.2.16.5 ASE_SPD.1.4C

The security problem definition shall describe the assumptions about the operational environment of the TOE.

6.2.17 ASE_OBJ.2 Security objectives

Dependencies satisfied by: ASE_SPD.1 Security problem definition

Developer action elements:

6.2.17.1 ASE_OBJ.2.1D

The developer shall provide a statement of security objectives.

6.2.17.2 ASE_OBJ.2.2D

The developer shall provide a security objectives rationale.

Content and presentation of evidence elements:

6.2.17.3 ASE_OBJ.2.1C

The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.

6.2.17.4 ASE_OBJ.2.2C

The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.

6.2.17.5 ASE_OBJ.2.3C

The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.

6.2.17.6 ASE_OBJ.2.4C

The security objectives rationale shall demonstrate that the security objectives counter all threats.

6.2.17.7 ASE_OBJ.2.5C

The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.

6.2.17.8 ASE_OBJ.2.6C

The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

6.2.18 ASE_ECD.1 Extended components definition

Dependencies satisfied by: No dependencies.

Developer action elements:

6.2.18.1 ASE_ECD.1.1D

The developer shall provide a statement of security requirements.

6.2.18.2 ASE_ECD.1.2D

The developer shall provide an extended components definition.

Content and presentation of evidence elements:

6.2.18.3 ASE_ECD.1.1C

The statement of security requirements shall identify all extended security requirements.

6.2.18.4 ASE_ECD.1.2C

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

6.2.18.5 ASE_ECD.1.3C

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

6.2.18.6 ASE_ECD.1.4C

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

6.2.18.7 ASE_ECD.1.5C

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

6.2.19 ASE_REQ.2 Derived security requirements

Dependencies satisfied by: ASE_OBJ.2 Security objectives

ASE_ECD.1 Extended components definition

Developer action elements:

6.2.19.1 ASE_REQ.2.1D

The developer shall provide a statement of security requirements.

6.2.19.2 ASE_REQ.2.2D

The developer shall provide a security requirements rationale.

Content and presentation of evidence elements:

6.2.19.3 ASE_REQ.2.1C

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

6.2.19.4 ASE_REQ.2.2C

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

6.2.19.5 ASE_REQ.2.3C

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

6.2.19.6 ASE_REQ.2.4C

All operations shall be performed correctly.

6.2.19.7 ASE_REQ.2.5C

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

6.2.19.8 ASE_REQ.2.6C

The security requirements rationale shall trace each SFR back to the security objectives for the TOE.

6.2.19.9 ASE_REQ.2.7C

The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.

6.2.19.10 ASE_REQ.2.8C

The security requirements rationale shall explain why the SARs were chosen.

6.2.19.11 ASE_REQ.2.9C

The statement of security requirements shall be internally consistent.

6.2.20 ASE_TSS.1 TOE summary specification

Dependencies satisfied by: ASE_INT.1 ST Introduction

ASE_REQ.2 Derived security requirements

ADV_FSP.4 Complete functional specification

Developer action elements:

6.2.20.1 ASE_TSS.1.1D

The developer shall provide a TOE summary specification.

Content and presentation of evidence elements:

6.2.20.2 ASE_TSS.1.1C

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

6.2.21 ATE_COV.2 Analysis of coverage

Dependencies satisfied by: ADV_FSP.4 Complete functional specification

ATE_FUN.1 Functional testing

Developer action elements:

6.2.21.1 ATE_COV.2.1D

The developer shall provide an analysis of the test coverage.

Content and presentation of evidence elements:

6.2.21.2 ATE_COV.2.1C

The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

6.2.21.3 ATE_COV.2.2C

The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

6.2.22 ATE_DPT.1 Testing: basic design

Dependencies satisfied by: ADV_ARC.1 Security Architecture Description

ADV_TDS.3 Basic modular design

ATE_FUN.1 Functional testing

Developer action elements:

6.2.22.1 ATE_DPT.1.1D

The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements:

6.2.22.2 ATE_DPT.1.1C

The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.

6.2.22.3 ATE_DPT.1.2C

The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

6.2.23 ATE_FUN.1 Functional testing

Dependencies satisfied by: ATE_COV.2 Analysis of coverage

Developer action elements:

6.2.23.1 ATE_FUN.1.1D

The developer shall test the TSF and document the results.

6.2.23.2 ATE_FUN.1.2D

The developer shall provide test documentation.

Content and presentation of evidence elements:

6.2.23.3 ATE_FUN.1.1C

The test documentation shall consist of test plans, expected test results and actual test results.

6.2.23.4 ATE_FUN.1.2C

The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

6.2.23.5 ATE_FUN.1.3C

The expected test results shall show the anticipated outputs from a successful execution of the tests.

6.2.23.6 ATE_FUN.1.4C

The actual test results shall be consistent with the expected test results.

6.2.24 ATE_IND.2 Independent testing - sample

Dependencies satisfied by: ADV_FSP.4 Complete functional specification

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

ATE_COV.2 Analysis of coverage

ATE_FUN.1 Functional testing

Developer action elements:

6.2.24.1 ATE_IND.2.1D

The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

6.2.24.2 ATE_IND.2.1C

The TOE shall be suitable for testing.

6.2.24.3 ATE_IND.2.2C

The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

6.2.25 AVA_VAN.5 Advanced methodical vulnerability analysis

Dependencies satisfied by: ADV_ARC.1 Security Architecture Description

ADV_FSP.4 Complete functional specification

ADV_TDS.3 Basic modular design

ADV_IMP.1 Implementation representation of the TSF

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative procedures

ATE_DPT.1 Testing: basic design

Developer action elements:

6.2.25.1 AVA_VAN.5.1D

The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

6.2.25.2 AVA_VAN.5.1C

The TOE shall be suitable for testing.

6.3 Security Functional Requirements Rationale

The following table shows that all SFRs contribute to at least one objective and all objectives are met at least by one SFR.

	O.AUDIT	O.AUTH	O.ALARM	O.ACCESSCONTROL	O.FLOW	O.CRYPTOOP	O.MACP
FAU_ARP.1			X				
FAU_GEN.1	X		X				
FAU_GEN.2	X		X				
FAU_SAA.1			X				
FAU_SAR.1	X						
FAU_SAR.2	X						
FCS_COP_EXT.1						X	
FDP_ACC.1/MANAGEMENT				X			
FDP_ACF.1/MANAGEMENT				X			
FDP_ACC.1/MAINTENANCE							X
FDP_ACF.1/MAINTENANCE							X

	O.AUDIT	O.AUTH	O.ALARM	O.ACCESSCONTROL	O.FLOW	O.CRYPTOOP	O.MACP
FDP_ETC.1							X
FDP_IFC.1					X		
FDP_IFF.1					X		
FDP_ITC.1							X
FIA_AFL.1		X					
FIA_ATD.1		X					
FIA_SOS.1		X					
FIA_UAU.2		X					
FIA_UID.2	X	X		X			
FMT_MSA.1				X			
FMT_MSA.3				X			
FMT_SMF.1				X			X
FMT_SMR.1				X			X

The following table shows how the SFRs satisfy the security objectives.

Objective	Rationale
O.AUDIT	The generation of audit records is performed by FAU_GEN.1. FAU_GEN.2 is the responsible for assigning the user identification provided by FIA_UID.2 to the records generated by identified users of the TOE. FAU_SAR.1 and FAU_SAR.2 provide the review capability to authorized users.
O.AUTH	The identification and authentication of administrative user that access through the management interface is conducted by FIA_UAU.2 and FIA_UID.2. The security attributes for each administrative user is maintained with FIA_ATD.1. The administrative user passwords follow the restrictions stated in FIA_SOS.1. The authentication attempts are controlled under FIA_AFL.1.
O.ALARM	FAU_GEN.1 and FAU_GEN.2 generate the audit records to be inspected by FAU_SAA.1 to detect potential violations. FAU_ARP.1 is the responsible of perform certain actions is case of detecting potential violations.

Objective	Rationale
O.ACCESSCONTROL	The access control policy for administrative users is performed by FDP_ACC.1/MANAGEMENT and FDP_ACF.1/MANAGEMENT. The user identity of the administrative users is provided by FIA_UID.2. The management of the access control policy security attributes is conducted by FMT_MSA.1 and FMT_MSA.3. FMT_SMF.1 provides the management functionality available for each role after the access control. The security roles are maintained by FMT_SMR.1.
O.FLOW	The dataflow control policy is conducted by FDP_IFC.1, FDP_IFF.1.
O.CRYPTOOP	The cryptographic operations invocation is conducted by FCS_COP_EXT.1.
O.MACP	The access control for Maintenance User is conducted by FDP_ACC.1/MAINTENANCE, FDP_ACF.1/MAINTENANCE. The accessible functionality after the access control for the Maintenance User is defined in FDP_ITC.1, FDP_ETC.1 and FMT_SMF.1. The TOE maintains the maintenance role with FMT_SMR.1.

6.4 Security Assurance Requirements Rationale

The overall security claim of this Security Target is aimed at ELA4+ with the augmentation of ALC_FLR.2 and AVA_VAN.5.

EAL4 is accepted as the suitable assurance level where TOE can be conformant. While the TOE will be connecting secure networks with public networks, it would be better to claim high attack potential and also in order to demonstrate the maintenance capability ALC_FLR.2 claimed.

All the dependencies and requirements of the selected assurance level and augmented components are satisfied during the life cycle of the TOE.

7. TOE SUMMARY SPECIFICATIONS

7.1 Audit (AUD)

Audit Alarms (AUD_ALR)

Automatic alarms will be generated when defined regular patterns are caught in the audit logs. According to the level of the pattern (either critical or non-critical) TOE will either put the TOE in passive mode or just send an information message to the administrators of the management console.

This functionality is satisfying the requirement FAU_ARP.1.

Audit Data (AUD_DAT)

The TOE generates audit logs according to a predefined policy. The recorded events in the audit logs are the included in FAU_GEN.1. Where applicable, the events will be associated to their subjects.

This functionality satisfies the requirements stated in FAU_GEN.1 and FAU_GEN.2 by generating audit logs.

Audit Analysis (AUD_ANL)

Audit logs will be automatically analyzed searching for predefined patterns and an alarm will be generated if any pattern is found.

This functionality satisfies the requirements stated in FAU_SAA.1 by collecting the audit logs from **vag-int** and **vag-ext** and then searching the logs against potential violations and generating alarms.

Audit Review (AUD_REV)

Audit logs can be reviewed via management interface and only the authorized administrative users can be able to review the audit logs.

This functionality satisfies the requirements stated in FAU_SAR.1 and FAU_SAR.2 by allowing only administrative users with sufficient access rights to review audit logs according to their role.

7.2 Cryptographic Operation Invocation (CRP)

The TOE Environment will encrypt/decrypt, sign/verify the data flow between **vag-int** and **vag-ext** by using OpenSSL algorithms upon invocation of the TOE. The cryptographic keys used for these algorithms are stored obfuscated for protection purposes.

This functionality satisfies the requirements stated in FCS_COP_EXT.1.

7.3 Data Protection (DPT)

DP Management Access Control (DPT_ACT)

Three types of administrative user are accessible to the TOE through the management interface. These three types are: administrator, manager and operator. Depending on their role, the will be able to access a certain set of management functions. The **management interface access control policy** is the responsible of providing access rights to administrative users taking into account the certain type of administrative user. The accessible functionality for each type of user is the following:

- **Operator:** This kind of users can open multiple sessions from different client hosts. They are able to perform the following actions:
  - Read system information.
  - Partially read configuration data.
  - Change its password.
  - Read audit logs.

- **Manager:** Managers can perform all the operations that an operator can, plus the following actions:
  - Read complete configuration data and modify partial configuration data.
  - Read administrative users list
  - Create partial backups.
  - Get list of available partial backups.

- **Administrator:** Administrator of the system is a single entity having full control over all available functionalities of the management interface. S/he can perform all the operations that a manager can, and additionally can perform the following actions:
  - Create/Modify/Delete administrative users.
  - Change the password of any administrative user.
  - Restore partial backups.

This functionality satisfies the requirements stated in FDP_ACC.1/MANAGEMENT and FDP_ACF.1/MANAGEMENT by enforcing access control policy for management interface.

DP Maintenance Access Control (DPT_ACM)

A Linux Shell access is provided for certain management purposes. This access is only available for the Maintenance User. This Maintenance User must perform an authentication and identification before conducting any action in the TOE. The responsibility of this authentication and identification belongs to the TOE environment (Debian Linux OS), which after this process will provide the user ID to the TOE. This user ID will be used by the TOE when exercising the **maintenance access control policy** to grant access to the Maintenance User to access the following functionality:

- Mount/Unmount USB devices.
- Install patches.

- Export audit logs.
- Export full backups.
- Restore full backups.
- Change its own password.
- Read license.

This functionality satisfies the requirements stated in FDP_ACC.1/MAINTENANCE and FDP_ACF.1/MAINTENANCE by enforcing access control policy for Linux Shell.

DP Flow Control (DPT_FCT)

The TOE provides a **data flow control policy** between the internal and the external network. This **data flow control policy** allows only permitted packets flow through the TOE.

This functionality satisfies the requirements FDP_IFC.1 and FDP_IFF.1.

DP Import (DPT_IMP)

Maintenance User can import patch files and full backups to the TOE according to the **maintenance access control policy**.

This functionality satisfies the requirements stated in FDP_ITC.1.

DP Export (DPT_EXP)

Maintenance User can export audit logs and full backups from the TOE according to the **maintenance access control policy**.

This functionality satisfies the requirements stated in FDP_ETC.1.

7.4 Identification and Authentication (IAU)

IA Authentication Failures (IAU_AUF)

The TOE provides login functionality in the management interface. The credentials of the administrative users are checked before granting access to the management interface accessible functionality. The TOE will disable the administrative user account if three (3) unsuccessful authentication attempts are reached. This functionality will limit the number of unsuccessful attempts, and therefore, it protects the management interface of the TOE against brute force attacks.

This functionality satisfies the requirements stated in FIA_AFL.1.

IA Password Quality (IAU_PQL)

Administrative users of the TOE can only choose passwords satisfying a certain quality metrics. During the modification of administrative user passwords, the TOE checks if the new password satisfies the password policy.

This functionality satisfies the requirements stated in FIA_SOS.1.

IA User Attributes (IAU_ATD)

The TOE controls, for each administrative user that access to the TOE through the management interface, its password and its role.

This functionality satisfies the requirements stated in FIA_ATD.1.

IA User Authentication (IAU_UAU)

Administrative users of the TOE are authenticated before conducting any action through the management interface.

This functionality satisfies the requirements stated in FIA_UAU.2.

IA User Identification (IAU_UID)

Administrative users of the TOE are identified before conducting any action through the management interface.

This functionality satisfies the requirements stated in FIA_UID.2

7.5 Security Management (SEM)

SM Security Attributes (SEM_SAT)

Only the administrator can modify the default value of administrative user role according to the management interface access control policy. The administrator is also able to modify the role of other administrative users.

This functionality satisfies the requirements stated in FMT_MSA.1 and FMT_MSA.3.

SM Functions (SEM_FUN)

The following management functions can be performed by TSF;

- Configuration Data management
- Administrative Users management
- Passwords management
- Partial Backups management
- Full Backups management
- Audit logs management
- Patch management
- USB devices management
- Read Licence

This functionality satisfies the requirements stated in FMT_SMF.1.

SM Roles (SEM_ROL)

TOE maintains four types of users that are named as administrator, manager, operator and maintenance. An administrative user can be administrator, manager or operator while the Maintenance User has the maintenance role.

This functionality satisfies the requirements stated in FMT_SMR.1.

Document Revision History

Revision No	Revision Reason	Date of Revision
0.1	First Draft	27/12/2010
0.2	Template Update	04/01/2011
0.3	Updated after remarks from ITSEF	12/01/2011
0.4	Updated with non-TOE environmental info details	07/02/2011
0.5	Release to the CB and ITSEF	10/02/2011
0.6	Updated according to the Observation Reports from the lab	24/05/2011
0.7	Updated after an internal control by the developer	15/06/2011
0.8	Adding abbreviations of TOE summary specifications	08/09/2011
0.9	Prepared for the submission to the evaluation facility	14/09/2011
0.9a	Updated after an internal control by the developer	10/10/2011
1.0	Updated according to the ORs from the ITSEF.	18/11/2011
1.1	Added missing updates for 1.0.	22/11/2011
1.1a	Updated according to the ORs from the ITSEF.	19/12/2011
1.1c	Updated according to the OR #	25/12/2011
1.2	Prepared for submission to the evaluation facility	04/01/2012
1.3	Updated according to the remarks from the evaluation facility	17/01/2012
1.4	Updated according to the response from the evaluation facility	26/01/2012
1.4b	Updated according to the response from the evaluation facility	03/02/2012
1.5	FDP_ACC and FDP_ACF is updated based on clarifications for access rights.	08/02/2012
1.5d	Updated according to the response from the evaluation facility	24/02/2012
1.6	Updated according to the response from the evaluation facility	28/02/2012
1.6a	Updated according to the response from the evaluation facility	02/03/2012
1.7	Updated according to the response from the evaluation facility	09/03/2012
1.8	Updated according to the response from the evaluation facility	08/05/2012
1.8b	Updated according to the comments from the evaluation facility	21/06/2012
1.8d	Updated according to the comments from the evaluation facility	30/07/2012
1.8e	Updated according to the comments from the evaluation facility	31/07/2012
1.8f	Updated according to the comments from the evaluation facility	01/08/2012
1.9	Updated according to the comments from the evaluation facility	02/08/2012
1.10	Updated according to the comments from the evaluation facility	06/08/2012