# Attivo BOTsink Solution

## Security Target

Version 1.3

## TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# ABBREVIATIONS

| Abbreviation | Description |
|---|---|
| AD | Active Directory |
| ACM | Attivo Central Manager |
| AWS | Amazon Web Services |
| CLI | Command Line Interface |
| GUI | Graphical User Interface |
| LDAP | Lightweight Directory Access Protocol |
| RDBMS | Relational DataBase Management System |
| SSH | Secure Shell |
| TSF | TOE Security Function |
| TSP | TOE Security Policy |
| UTC | Coordinated Universal Time (French: Temps universel coordonné) |
| VM | Virtual Machine |

# DEFINITIONS

| Definition | Description |
|---|---|
| ACM | The physical Attivo Central Manager appliance |
| AD | Microsoft Windows directory service that facilitates working with interconnected, complex and different network resources in a unified manner |
| AWS | A subsidiary of Amazon.com, which offers a suite of cloud computing services that make up an on-demand computing platform |
| vACM | The virtual ACM appliance |
| xACM | Physical ACM or virtual vACM |
| BOTsink | The physical appliances (e.g. 3200/5100) not including ACM |
| vBOTsink | The virtual appliances (e.g. vBOTsink VMware, vBOTsink AWS) |
| xBOTsink | Physical BOTsink or virtual vBOTsink |
| CLI | Means of interacting with a computer program where the user (or client) issues commands to the program in the form of successive lines of text (command lines) |
| GUI | Allows users to interact with electronic devices through graphical icons and visual indicators |
| LDAP | An open, vendor-neutral, industry standard application protocol for accessing and maintaining distributed directory information services over an Internet Protocol (IP) network |
| Management Domain | Either the host Operating System on BOTsink/ACM or the Management Virtual Machine on vBOTsink/vACM |
| Endpoint agent | Attivo software for communication with xBOTsink |
| MySQL | Open-source relational database management system (RDBMS) |
| Hardware virtualization | Virtualization of computers as complete hardware platforms, certain logical abstractions of their componentry, or only the functionality required to run various operating systems |
| Pivot | A threat originating from newly compromised host. A threat agent is said to have pivoted, when it compromises a host and then proceeds to attack from that compromised host |

| Definition | Description |
|---|---|
| RDBMS | Database management system (DBMS) is based on the relational model (RM), which is an approach to managing data using a structure and language consistent with first-order predicate logic, which is a collection of formal systems used in mathematics, philosophy, linguistics, and computer science |
| SSH | Cryptographic (encrypted) network protocol to allow remote login and other network services to operate securely over an unsecured network |
| UTC | Coordinated Universal Time, is the primary time standard by which the world regulates clocks and time |
| Virtual machine | An emulation of a particular computer system |
| Virtualization | The act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, operating systems, storage devices, and computer network resources |

# 1. ST Introduction (ASE_INT)

## 1.1. ST and TOE References

The following table identifies the Security Target (ST).

| Item | Identification |
|---|---|
| ST title | Attivo BOTsink Solution Security Target |
| ST version | 1.3 |
| ST author | Attivo Networks Inc. |

The following table identifies the Target of Evaluation (TOE).

| Item | Identification |
|---|---|
| TOE name | Attivo BOTsink Solution |
| TOE hardware models/versions | BOTsink appliance models: 3200, 5100<br>ACM appliance version 200 |
| TOE software versions | vBOTsink version 3.3 for Vmware<br>BOTsink and Endpoint version 3.3<br>ACM version 3.3 |

The following table identifies common references for the ST and the TOE.

| Item | Identification |
|---|---|
| CC Version | 3.1 Revision 4 |
| Assurance level | EAL2 augmented with ALC_FLR.1 |
| Protection Profile | None |

## 1.2. TOE Overview

The intention of the Attivo BOTsink Solution is to close security infrastructure gaps. Attivo BOTsink is an advanced decoy and deception solution, which detects network breaches and can stop threats that have bypassed prevention security systems from further propagation in the network. Additionally, this solution can be configured for analyzing suspect content submitted by users and partner security devices. Both virtualized and physical instances of Attivo BOTsink are available. This means that the Attivo BOTsink Solution can be deployed in networks, datacenters and on the cloud, see figure 1.

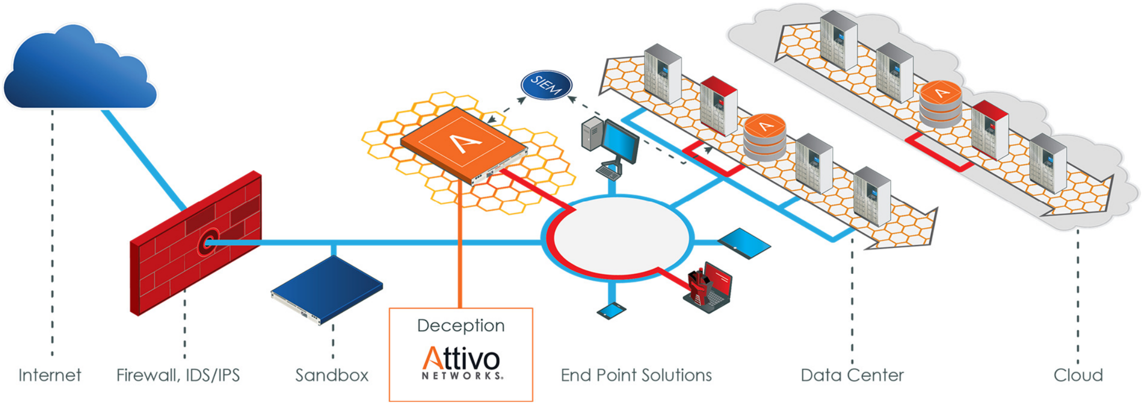

**Figure 1: Security Infrastructure Solutions**

The BOTsink solution basically consists of a network of self-sustaining virtual machines running on various operating systems. A set of configurable network services run on these virtual machines. The user can deploy these virtual machines on the required subnets.

BOTsink lures BOTs and APTs scanning for valuable corporate assets to target its virtual machines. Thus, Attivo BOTsink leaves a wide footprint across the enterprise to detect, engage, and defend against BOTs and APTs.

An implementation of the architecture including all parts of the system could look like figure 2 below:

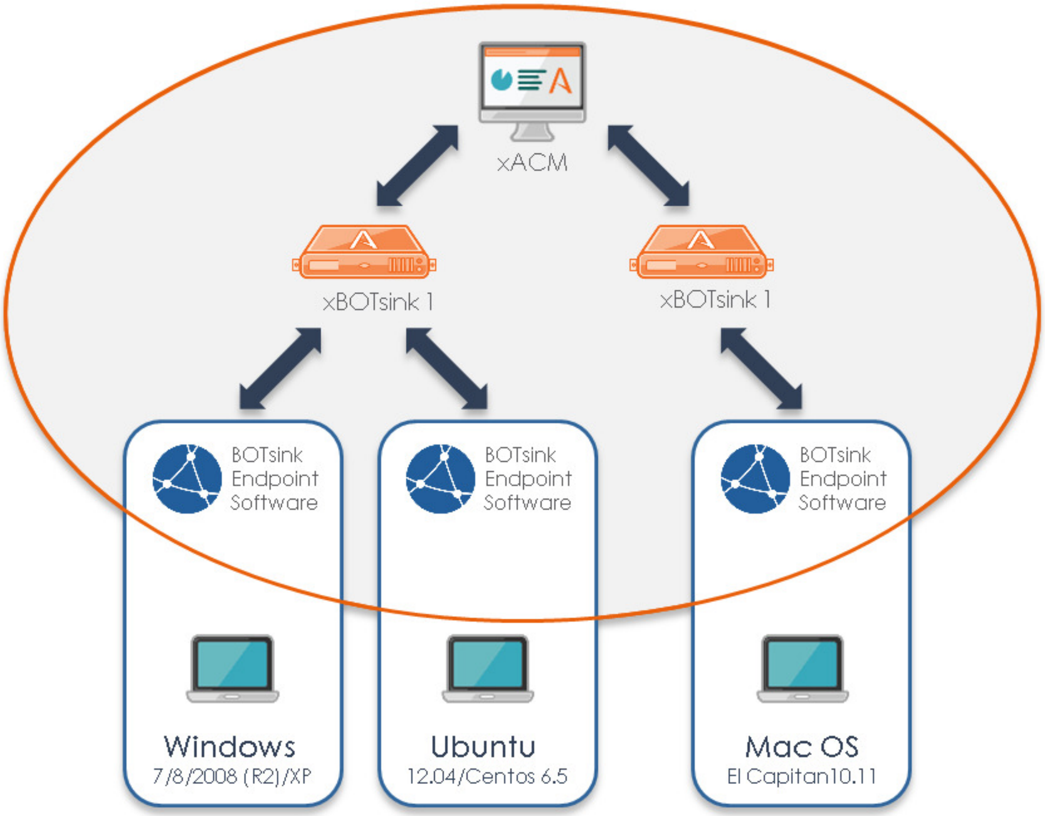

**Figure 2: Architecture implementation**

The xACM is designed to manage several xBOTsinks in a centralized manner. xACMs cannot be managed by other xACMs, and each xBOTsink can be programmed to use one and only one xACM. The endpoint software allows enabled hosts running supported versions of Windows, Linux or MacOS to connect back to xBOTsink to get configuration information and submit relevant information for centralized processing and threat intelligence. Each endpoint software instance is pre-configured during installation to talk to a particular xBOTsink.

The hosted virtual machines in xBOTsink are either configured to be vulnerable honeypots that can be easily discovered by threat agents that are active in a network or used for analyzing suspect content submitted by users and partner security devices.

Additionally, the BOTsink 5100 and ACM 200 provide power redundancy to the physical boxes (2 power connectors), and also disk redundancy (RAID mirrors) in case disks fail.

### 1.2.1. Usage and major security features of the TOE

Attivo BOTsink provides an additional layer of security in case the perimeter security systems are bypassed. Real-time notifications from BOTsink alert the user about the possible breaches when the attacks are in their nascent stages and before any data is exfiltrated. Thus, Attivo BOTsink complements and augments the network Intrusion Prevention System (IPS) and Firewalls.

### 1.2.2. TOE type

The Attivo BOTsink Solution (TOE) is categorized as a decoy and deception solution, which detects network breaches and can deceive attackers into revealing themselves.

### 1.2.3. Required non-TOE software and hardware

TOE requires one of the following endpoint OS:

- Windows 7/8/2008(R2)/XP
- Ubuntu 12.04/CentOS 6.5
- MacOS El Capitan 10.11

TOE requires the following hardware requirements for the endpoint OS:

- Window 7 64-bit has the following specifications:
  - >= 1 GHz 64-bit CPU
  - >= 2 GB RAM
  - 20 GB HDD
- Ubuntu has the following specifications:
  - >= 700 MHz Intel Celeron
  - 512 MB RAM
  - 5GB HDD
- The Mac mini (for the Mac endpoint) has the following specifications:
  - 1.4 GHz Intel i5
  - 4 GB RAM
  - 500 GB HDD

The TOE requires the following hardware requirements for vBOTsink for VMware:

| **vBOTsink** | Management VM |
|---|---|
| **Number of cores requires/ Recommended** | 2/4 |
| **Memory required/ Recommended** | 16GB/32GB |
| **Disc Space required/ Recommended** | 150GB/1TB |

Additionally, the BOTsink Manager, which is a web application hosted on the management VM, is accessed by a browser. The latest versions of the following browsers are supported to access the BOTsink Manager:

- Internet Explorer
- Firefox
- Chrome
- Safari

## 1.3. TOE Description

### 1.3.1. Physical scope

The TOE is a hardware, firmware and software solution comprised of the components described below:

- BOTsink appliance model 3200

- BOTsink appliance model 5100
- vBOTsink applianceis physically installed on platforms that support Vmware
- ACM appliance
- BOTsink endpoint software agent is physically installed on hosts running non-TOE OS

The supporting guidance documents are:

1. Administrator Guide for FIPS and Common Criteria Certification, v. 1.0
2. Attivo BOTsink® Software version 3.1.1, Deployment Scenarios Guide Revision A
3. Attivo BOTsink® Software version 3.1.1, Installation Guide for VMware Revision A
4. Attivo BOTsink® Software version 3.3.3, IRES Cheat Sheet Revision A
5. Attivo BOTsink® Software version 3.3.3, Central Manager User Guide Revision A
6. Attivo BOTsink® Software version 3.3.3, User Guide Revision D
7. Attivo vBOTsink-AWS Software version 3.2.1, User Guide Revision A
8. Attivo vBOTsink-VMware Requirement details
9. ACM Alerts Design Document, v.0.3
10. ACM Integrated Help document
11. BOTsink Integrated Help document
12. VMware Integrated Help document

#### 1.3.1.1. TOE Evaluated Configuration

The TOE is defined as Attivo BOTsink Solution with TOE hardware models/versions and TOE software versions as defined in the TOE references table in section 1.1:

| Component | Version number |
|---|---|
| BOTsink appliances | • Model 3200<br>• Model 5100 |
| vBOTsink for Vmware | Version 3.3 |
| ACM appliance | Version 200 |
| Software - ACM | Version 3.3 |
| Software - BOTsink and Endpoint | Version 3.3 |

### 1.3.2. Logical scope

The TOE is comprised of several security features. Each of the security features identified consists of several security functionalities, as identified below.

1. Security Audit
2. Identification and Authentication
3. Security Management
4. Protection of the TSF
5. TOE Access
6. Cryptographic Support
7. Network Isolation

These features are described in more detail in the subsections below.

#### 1.3.2.1. Security Audit

The Attivo BOTsink Solution provides extensive auditing capabilities. The TOE generates a comprehensive set of audit logs that identify specific TOE operations. For each event, the TOE records the date and time of each event, the type of event, the subject identity and the outcome of the event. The TOE can audit several events, including events related to identification, authentication, and administration (system activity triggered by the internal TOE activities).

### 1.3.2.2. Identification and Authentication

The TOE provides authentication services for local users and AD users wishing to connect to the GUI interface. AD users are granted access if and only if LDAP authentication succeeds against the AD server specified by the administrator(s). Local users do not require network communication and are authenticated against a local user database. A user can be an authorized administrator or a non-privileged user. The non-privileged users cannot configure the device and are limited to reading information such as security events and analysis reports, and they cannot see current device configuration.

The TOE provides authentication services for administrative users wishing to connect to the TOEs secure CLI administrator interface on the device console, the serial console or SSH interfaces.

The TOE requires authorized administrators to authenticate prior to being granted access to any of the management functionality. For AD based GUI users, the password restrictions are determined by the enterprise. For local GUI users, the TOE can be configured to require a minimum password length of 8 characters as well as mandatory password complexity rules. For CLI users, standard password guidelines apply, meaning no complexity enforcement.

### 1.3.2.3. Security Management

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. All TOE administration occurs through a secure SSH/HTTPS session, or a local (or serial) console connection.

For xACM to start managing xBOTsink, the device name and the IP address of the xBOTsink would need to be specified in xACM management interface. In addition, the xBOTsink must be configured with the device name and the IP address of the xACM. The communication between xACM and xBOTsinks managed by the xACM is over TLS 1.2 with mutual authentication. This channel carries remote administration and centralized policy information that gets applied to the xBOTsinks. Each xBOTsink reports security and network events back to the xACM configured for it. This communication is also over TLS 1.2 with mutual authentication.

The endpoint agent communicates to xBOTsink to send installation status and other endpoint information. Each endpoint agent is configured to talk to an xBOTsink management domain IP address or DNS name. The communication between xBOTsink and endpoint agents is over TLS with mutual authentication. The endpoint agent is seeded with its client side certificate and the trusted server certificate. Once the HTTPS session is established, a token must be presented by endpoint agent before other communication is successful.

The TOE provides the ability to securely manage TOE updates. All firmware updates are digitally signed and encrypted.

### 1.3.2.4. Protection of the TSF

The TOE protects against interference and tampering by untrusted subjects by implementing identification, authentication, and access controls to limit configuration to authorized administrators.

The TOE internally maintains the date and time. This date and time is used as the timestamp that is applied to audit records generated by the TOE.

### 1.3.2.5. TOE Access

When a user SSH/Web GUI session is initially established, the TOE displays warning banners. In addition, when a user establishes a local console session or a serial console session, the TOE displays a warning banner. The banners are used to provide any information deemed necessary for the user. After a period of inactivity, user sessions will be terminated, requiring users to re-authenticate.

### 1.3.2.6. Cryptographic Support

The TOE provides cryptographic support for the following features.

- Trusted path/secure channels of communication to GUI interface (HTTPS).
- TLS channels between endpoint agent and BOTsink.
- TLS channels between xACM and xBOTsink.
- CLI remote access is over secure channel (SSH).
- The TOE can communicate securely (TLS) with email server IMAPS/SMTPS for email based content submissions.
- The TOE communicates securely (TLS) to AD server over LDAPS when AD is configured.
- Updates to TOE are digitally signed and AES encrypted.
- The TOE provides FIPS approved cryptographic algorithms implemented in a FIPS 140-2 compliant crypto-module to support SSH and TLS/HTTPS usage.

### 1.3.2.7. Network Isolation

Because the honeypot virtual machines in xBOTsink are vulnerable, they are isolated from management domain and other parts of the deployment network for mitigating security risk from a possible pivot from these machines. All outbound communication from the honeypot virtual machines is redirected to Sinkhole in xBOTsink, or dropped by firewall rules.

The Sinkhole may be configured to proxy the connection to the Internet for deeper understanding of threat intent. If the Sinkhole is not configured to proxy (default), then all outbound connections terminate in Sinkhole.

When a virtual machine is used for suspect content analysis purposes, it never joins the deployment network.

All non-management virtual machines in xBOTsink send information back to management domain of deployment network through a dedicated management interface.

## 1.4. Notations and Formatting

The notations and formatting used in this ST are consistent with version 3.1 Revision 4 of the Common Criteria (CC).

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**. Deleted words are denoted by ~~strike-through text~~.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections are denoted by *italicized* text in square brackets, [*Selection value*].

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignment is indicated by showing the value with bold face in square brackets, [**Assignment_value**].

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing the iteration number in parenthesis following the component identifier, (iteration_number).

**Assets:** Assets to be protected by the TOE are given names beginning with "AS." – e.g. AS.CLASSIFIED_INFO.

**Assumptions:** TOE security environment assumptions are given names beginning with "A."- e.g., A.Security_Procedures.

**Threats:** Threat agents are given names beginning with "TA." – e.g., TA.User. Threats to the TOE are given names beginning with "TT." – e.g., TT.Filter_Fails. TOE security environment threats are given names beginning with "TE."-- e.g., TE.Crypto_Fails.

**Policies:** TOE security environment policies are given names beginning with "P."—e.g., P.Information_AC.

**Objectives:** Security objectives for the TOE and the TOE environment are given names beginning with "O." and "OE.", respectively, - e.g., O.Filter-msg and OE.Clearance.

# 2. CC Conformance Claim (ASE_CCL)

This TOE and ST are conformant with the following specifications.

| Item | Identification |
|---|---|
| CC Part 2 | Security functional components, September 2012, Version 3.1, Revision 4, extended |
| CC Part 3 | Security assurance components, September 2012, Version 3.1, Revision 4, conformant, EAL2 augmented with ALC_FLR.1 |
| Assurance level | EAL2 augmented with ALC_FLR.1 |
| Protection Profile | None |
| Package conformance | None |
| Extended SFRs | FRU_RSA_EXT.1 |

# 3. Security Problem Definition (ASE_SPD)

## 3.1. Threats to security

### 3.1.1. Assets

The assets that TOE shall protect as specified in this Security Target are described in the following.

| Assets | Description |
|---|---|
| AS.DATA | Authentication data, audit records, threat intelligence information and other sensitive or security functional data. |
| AS.SERVICE | Services running on physical instance(s) of BOTsink, services running on virtualized instance(s) of BOTsink, and ACM services. |
| AS.DEPLOY_NETWORK | System services, resources and information on the deployment network. |

### 3.1.2. Threat Agents

| Threat Agents | Description |
|---|---|
| TA.ATTACKER | A person/company or process with skills and resources to mislead the system in any way necessary to misuse services and data and prevent the system from operating. |
| TA.ADMIN | Authorized person/process that performs installation and configuration/setup of the TOE to ensure that the TOE operates according to the needs of the deployment network system. |

### 3.1.3. Identification of Threats

#### 3.1.3.1. Threats to the TOE

| **Threats to the TOE** | **Description** |
|---|---|
| TT.ADMIN_ERROR | The TOE may be incorrectly configured that may result in the TOE's acquisition of ineffective security mechanisms. |
| Threat agent: | TA.ADMIN |
| Assets: | AS.DATA, AS.SERVICE, AS.DEPLOY_NETWORK |
| Attack method: | During operation, the administrator unintentionally configures the TOE incorrectly, making the TOE inoperable or resulting in ineffective security mechanisms. |
| | |
| TT.ADMIN_EXPLOIT | A person/company may gain access to an administrator account. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA, AS.SERVICE |
| Attack method: | A person/company gains unauthorized TOE access through local console login, remote SSH login, remote GUI login, or serial login. |
| | |
| TT.AUDIT_COMPROMISE | A person/company may modify or remove audit records to mask actions in the past or prevent logging of actions in the future. |
| Threat agent: | TA.ATTACKER |

| Threats to the TOE | Description |
|---|---|
| Assets: | AS.DATA |
| Attack method: | A person/company uses hacking methods to exploit weakness in the TOE. |
| | |
| TT.CRYPTO_COMPROMISE | An attacker may compromise the data protected by the cryptographic mechanisms. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA |
| Attack method: | An attacker causes key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed/modified/deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms. |
| | |
| TT.EAVESDROPPING | Eavesdropping of the communication between physically separated parts of the TOE. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA |
| Attack method: | An unauthorized person with no physical access to TOE is eavesdropping on the communication between the TOE parts to intercept data. |
| | |
| TT.FLAWED_DESIGN | Unintentional or intentional errors in requirements specification or design of the TOE may occur. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA, AS.SERVICE, AS.DEPLOY_NETWORK |
| Attack method: | Errors in requirements specification or design of the TOE lead to flaws to be exploited by a malicious user or program. |
| | |
| TT.NETWORK_INTRUSION | A person/company may compromise a honeypot host (virtual machines in xBOTsink) to use it as a foothold to penetrate further into the TOE. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA, AS.SERVICE |
| Attack method: | A person/company uses hacking methods to exploit weakness in the TOE. |
| | |
| TT.PHYSICAL_TAMPER | The TOE may be subject to physical attack that may compromise TOE resources, including storage media of BOTsink or ACM, to discover sensitive security data that may then be used in an attack against the TOE. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA, AS.SERVICE |
| Attack method: | A person/company tampers with physical appliances or disks, by:<br>• insertion of unauthorized USB media<br>• removal or insertion of disks<br>• removal of enclosure |
| | |
| TT.REPLAY | The TOE may be subject to a replay attack that may compromise TOE resources. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA |

| Threats to the TOE | Description |
|---|---|
| Attack method: | An attacker manages to disclose data within packet flows transmitted and received by the TOE over an untrusted network. |
| | |
| TT.RESOURCE_DRAIN | The TOE may be subject to a DOS attack that may throttle the user services availability. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.SERVICE, AS.DEPLOY_NETWORK |
| Attack method: | An attacker claims network, processing or storage resources to such a degree that legitimate users cannot access or use services of the TOE. |
| | |
| TT.SPOOFING | An attacker may misrepresent itself as part of the TOE to get authentication data. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA |
| Attack method: | An attacker gains unauthorized TOE access through TLS certificates used in xBOTsink, xACM and endpoint agents. |
| | |
| TT.UNATTENDED_ SESSION | The TOE may be subject to a session attack that may compromise TOE resources. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA, AS.SERVICE |
| Attack method: | An attacker gains unauthorized TOE access through an unattended logon session. |
| | |
| TT.UNAUTHORIZED_UPDATE | A malicious party attempts to supply an update to the product that may compromise the security features of the TOE. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA, AS.SERVICE, AS.DEPLOY_NETWORK |
| Attack method: | During operation, the maintenance personal unintentionally installs a wrong SW or FW version in the TOE; making the TOE inoperable or SW/FW updates concerning security is missed. |
| | |
| TT.UNDETECTED_ACTIONS | Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated. |
| Threat agent: | TA.ATTACKER |
| Assets: | AS.DATA, AS.SERVICE, AS.DEPLOY_NETWORK |
| Attack method: | A malicious party attempts to establish a security association that leads to an unauthorized peer making itself a part of the TOE. |

### 3.1.3.2. Threats to the TOE environment

| Threats to the TOE environment | Description |
|---|---|
| TE.ADMIN_FAIL | The administrator may fail to perform functions essential to the security. |
| Threat agent: | TA.ADMIN |
| Assets: | AS.DEPLOY_NETWORK |
| Attack method: | The administrator fails to or forgets to update the TOE with security patches. |

## 3.2. Organizational security Policies

| Organizational security Policies | Description |
|---|---|
| P.ACCESS_BANNER | The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the system. |
| P.ACCOUNTABILITY | The authorized users of the TOE shall be held accountable for their actions within the TOE. |
| P.ADMIN_ACCESS | An authorized administrator must manage the TOE securely. |
| P.CRYPTOGRAPHIC | The TOE shall provide cryptographic functions for its own use, including encryption/decryption operations. |

## 3.3. Assumptions

The following conditions are assumed to exist in the operational environment.

| Assumptions | Description |
|---|---|
| A.NO_GENERAL_PURPOSE | There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. |
| A.PHYSICAL | The TOE shall presumably be located in physically secure environment that can be accessed only by the authorized administrators. |
| A.TRUSTED_ADMIN | The administrators of the TOE shall not have any malicious intention, shall receive proper training on the TOE management, and shall follow the administrator guidelines. |

# 4. Security objectives (ASE_OBJ)

## 4.1. TOE Security Objectives

| Security Objectives | Description |
|---|---|
| O.ACCESS | The TOE will provide mechanisms that control a user's logical access to the TOE and to explicitly deny access to specific users when appropriate. |
| O.AUDIT | The TOE shall record and maintain security-related events associated with users in order to enable tracing of responsibilities for security-related acts, and shall provide means to review the recorded data. |
| O.AUTH_COMM | The TOE will provide a means to ensure that users are not communicating with some other entity pretending to be the TOE, and that the TOE is communicating with an authorized IT entity and not some other entity pretending to be an authorized IT entity. |
| O.CHANGE_ MANAGEMENT | The configuration of, and all changes to, the TOE and its development evidence will be analyzed, tracked, and controlled throughout the TOE's development. |
| O.CRYPTOGRAPHY | The TOE shall provide cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of TSF data that is transmitted between physically separated portions of the TOE. |
| O.DISPLAY_BANNER | The TOE will display an advisory warning regarding use of the TOE. |
| O.MANAGE | The TOE shall provide means for the authorized administrator of the TOE to efficiently manage the TOE in a secure manner, including updating the TOE with security patches. |
| O.MEDIATE | The TOE must protect data in accordance with its security policy. |
| O.NETWORK_FILTERING | The TOE will provide the means to filter network packets. |
| O.REPLAY_DETECTION | The TOE must provide a means to detect that a packet flow transmitted to the TOE has not been copied by an eavesdropper and retransmitted to the TOE. |
| O.RESOURCE | The TOE shall provide mechanisms that mitigate attempts to exhaust the resources provided by the TOE. |
| O.SELF_PROTECTION | The TSF will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure through its own interfaces. |
| O.TIME_STAMPS | The TOE shall provide reliable time stamps and the capability for the administrator to set the timesource used for these time stamps, in that the administrator can configure the TOE to synchronize its clocks with NTP servers. |
| O.SOUND_DESIGN | The design of the TOE will be the result of sound design principles and techniques. The design of the TOE, as well as the design principles and techniques, are adequately and accurately documented. |

| Security Objectives | Description |
|---|---|
| O.THOROUGH_ FUNCTIONAL_ TESTING | The TOE will undergo appropriate security functional testing that demonstrates the TSF satisfies the security functional requirements. |
| O.VERIFIABLE_UPDATES | The TOE will provide the capability to help ensure that any updates to the TOE can be verified to be unaltered from a trusted source. |

## 4.2. Operational Environment Security Objectives

| Security Objectives | Description |
|---|---|
| OE.CRYPTANALYTIC | Cryptographic methods used in the IT environment shall be interoperable with the TOE, should be FIPS 140-2 validated and should be resistant to cryptanalytic attacks. |
| OE.NO_GENERAL_PURPOSE | There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. |
| OE.PHYSICAL | The TOE shall be located in physically secure environment that can be accessed only by the authorized administrator. |
| OE.TRUSTED_ADMIN | The administrators of the TOE shall not have any malicious intention, shall receive proper training on the TOE management, and shall follow the administrator guidelines. |

## 4.3. Security Objectives Rationale

The following tracing shows how the security objectives trace back to threats, organizational security policies (OSP) and assumptions defined by the SPD.

| **Threats/ Policies/ Assumptions**<br>**Objectives** | TT.ADMIN_ERROR | TT.ADMIN_EXPLOIT | TT.AUDIT_COMPROMISE | TT.CRYPTO_COMPROMISE | TT.EAVESDROPPING | TT.FLAWED_DESIGN | TT.NETWORK_INTRUSION | TT.PHYSICAL_TAMPER | TT.REPLAY | TT.RESOURCE_DRAIN | TT.SPOOFING | TT.UNATTENDED_ SESSION | TT.UNAUTHORIZED_UPDATE | TT.UNDETECTED_ACTIONS | TE.ADMIN_FAIL | P.ACCESS_BANNER | P. ACCOUNTABILITY | P.ADMIN_ACCESS | P.CRYPTOGRAPHIC | A.NO_GENERAL_PURPOSE | A.PHYSICAL | A.TRUSTED_ADMIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **TOE Security Objectives** | | | | | | | | | | | | | | | | | | | | | | |
| O.ACCESS | | X | | | X | | X | | | | X | X | | | | | X | | | | | |
| O.AUDIT | | | | | | | X | | | | | | | X | | | X | | | | | |
| O.AUTH_COMM | | | | | | | | | | | X | | | X | | | | | | | | |
| O.CHANGE_ MANAGEMENT | | | | | | X | | | | | | | | | | | | | | | | |
| O.CRYPTO-GRAPHY | | | | | X | | | | X | | | | | X | | | | | X | | | |
| O.DISPLAY_ BANNER | | | | | | | | | | | | | | | | X | | | | | | |
| O.MANAGE | X | X | | | X | | X | | | | | | | X | X | | | X | | | | |
| O.MEDIATE | | | | | X | | | | | | | | | | | | | | | | | |
| O.NETWORK_ FILTERING | | | | | | | | | | X | | | | | | | | | | | | |
| O.REPLAY_ | | | | | | | | | X | | | | | | | | | | | | | |

| **Threats/ Policies/ Assumptions**<br>**Objectives** | TT.ADMIN_ERROR | TT.ADMIN_EXPLOIT | TT.AUDIT_COMPROMISE | TT.CRYPTO_COMPROMISE | TT.EAVESDROPPING | TT.FLAWED_DESIGN | TT.NETWORK_INTRUSION | TT.PHYSICAL_TAMPER | TT.REPLAY | TT.RESOURCE_DRAIN | TT.SPOOFING | TT.UNATTENDED_ SESSION | TT.UNAUTHORIZED_UPDATE | TT.UNDETECTED_ACTIONS | TE.ADMIN_FAIL | P.ACCESS_BANNER | P. ACCOUNTABILITY | P.ADMIN_ACCESS | P.CRYPTOGRAPHIC | A.NO_GENERAL_PURPOSE | A.PHYSICAL | A.TRUSTED_ADMIN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DETECTION | | | | | | | | | | | | | | | | | | | | | | |
| O.RESOURCE | | | | | | | | | | X | | | | | | | | | | | | |
| O.SELF_ PROTECTION | | | X | X | | | X | X | | | | | | | | | | | | | | |
| O.SOUND_ DESIGN | | | | | | X | | | | | | | | | | | | | | | | |
| O.TIME_STAMPS | | | | | | | | | | | | | | | | | X | | | | | |
| O.THOROUGH_ FUNCTIONAL_ TESTING | X | X | X | X | X | | X | X | X | X | X | X | X | X | X | X | X | X | X | X | | |
| O.VERIFIABLE_ UPDATES | | | | | | | | | | | | | X | | | | | | | | | |
| **Operational Environment Security Objectives** | | | | | | | | | | | | | | | | | | | | | | |
| OE.CRYPT-ANALYTIC | | | | X | | | | | | | | | | | | | | | | | | |
| OE.NO_ GENERAL_ PURPOSE | | | | | | | | | | | | | | | | | | | | X | | |
| OE.PHYSICAL | | | | | | | | X | | | | | | | | | | | | | X | |
| OE.TRUSTED_ ADMIN | X | X | | | | | X | | | | | | | | X | | X | X | | | | X |

**Table 1: Mapping of Objectives to Threats, Policies and Assumptions**

The following justification demonstrates that the tracing of the security objectives to threats, OSPs and assumptions is effective, and all the given threats are countered, and all the given OSPs are enforced, and all the given assumptions are upheld.

| Threat/OSP/ Assumption | Security Objective Rationale |
|---|---|
| TT.ADMIN_ERROR | O.MANAGE provides administrators the capability to view and manage configuration settings.<br>OE.TRUSTED_ADMIN ensures that the administrators are non-hostile and are trained to appropriately manage and administer the TOE.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.ADMIN_EXPLOIT | O.ACCESS includes mechanisms to authenticate TOE users and place controls on user sessions.<br>O.MANAGE restricts access to administrative functions and management of TSF data to the administrator.<br>OE.TRUSTED_ADMIN ensures that the TOE administrators have guidance that instructs them how to administer the TOE in a |

| | |
|---|---|
| | secure manner.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.AUDIT_ COMPROMISE | O.SELF_PROTECTION ensures that the TSF can protect itself from users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the audit trail.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.CRYPTO_ COMPROMISE | O.SELF_PROTECTION ensures that the TSF can protect itself from malicious users. If the TSF could not maintain and control its domain of execution, it could not be trusted to control access to the resources under its control, which includes the cryptographic data and executable code.<br>OE.CRYPTANALYTIC ensures that the cryptographic methods used in the IT environment are interoperable with the mechanisms provided by the TOE. The IT environment's cryptographic methods should be independently validated to be FIPS 140-2 compliant.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.EAVESDROPPING | O.ACCESS provides the means to identify and authenticate the TOE user. The correct identity of the user is the basis for any decision of the TOE about an attempt of a user to access data.<br>O.CRYPTOGRAPHY requires the TOE to implement cryptographic services to provide confidentiality and integrity protection of data while in transit between parts of the TOE.<br>O.MANAGE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc. to the administrator. This feature ensures that no other user can modify the information flow policy to bypass the intended TOE security policy.<br>O.MEDIATE ensures that all accesses to data are subject to mediation. The TOE requires successful authentication to the TOE prior to gaining access to any content. By implementing strong authentication to gain access to these services, an attacker's opportunity to conduct an eavesdropping attack successfully is greatly reduced. The TSF will ensure that all configured enforcement functions (authentication, access control rules, etc.) must be invoked prior to allowing a user to gain access to TOE or TOE mediated services.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.FLAWED_ DESIGN | O.CHANGE_MANAGEMENT requires the developer to provide control of the changes made to the TOE's design. This includes controlling physical access to the TOE's development area, and having an automated configuration management system that ensures changes made to the TOE go through an approval process and only those persons that are authorized can make changes to the TOE's design and its documentation.<br>O.SOUND_DESIGN requires that the TOE be developed using sound engineering principles. By accurately and completely |

| | |
|---|---|
| | documenting the design of the security mechanisms in the TOE, the design of the TOE can be better understood, which increases the chances that design errors will be discovered. |
| TT.NETWORK_ INTRUSION | O.ACCESS includes mechanisms to authenticate TOE administrators and place controls on administrator sessions.<br>O.AUDIT provides the TOE the capability to detect and create records of security-relevant events associated with users.<br>O.MANAGE restricts the ability to modify the security attributes associated with access control rules, access to authenticated and unauthenticated services, etc. to the administrator. These objectives ensure that no other user can modify the information flow policy to bypass the intended TSP.<br>O.SELF_PROTECTION ensures that the TOE will have adequate protection from external sources and that all TSP functions are invoked.<br>OE.TRUSTED_ADMIN ensures that the TOE administrators have guidance that instructs them how to administer the TOE in a secure manner.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.PHYSICAL_ TAMPER | O.SELF_PROTECTION ensures that TOE provides a mechanism that detects the exposure of the internal TOE components. The TSF self-tests required by this objective ensure that the TOE's hardware is operating correctly, and the software and TSF data have not been corrupted by means other than exposing the internal components (e.g., electromagnetic interference).<br>OE.PHYSICAL provides for the physical protection of the TOE hardware and software.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.REPLAY | O.CRYPTOGRAPHY requires the TOE to implement cryptographic services to provide confidentiality and integrity protection of TLS/HTTPS sessions.<br>O.REPLAY_DETECTION ensures that a packet flow transmitted to the TOE has not been copied by an eavesdropper and retransmitted to the TOE.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.RESOURCE_ DRAIN | O.NETWORK_FILTERING ensures that the TOE filters network packets.<br>O.RESOURCE shall provide mechanisms that mitigate attempts to exhaust the resources provided by the TOE.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.SPOOFING | O.ACCESS controls the logical access to the TOE and its resources. By constraining how authorized users can access the TOE, and by mandating the type and strength of the authentication mechanism, this objective helps mitigate the possibility of a user attempting to login and masquerading as an authorized user. |

| | |
|---|---|
| | O.AUTH_COMM ensures that TOE identifies and authenticates all peers prior to communicating with that peer.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.UNATTENDED_ SESSION | O.ACCESS includes mechanisms that place controls on SSH/Web GUI/Console sessions. The sessions are dropped after a defined time period of inactivity. Dropping the connection of a session (after the defined time period) reduces the risk of someone accessing the TOE device where the session was established, thus gaining unauthorized access to the session.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.UNAUTHORIZED_ UPDATE | O.VERIFIABLE_UPDATES: To reduce the potential that an update might contain malicious or unintended features, the TOE is expected to provide mechanisms that serve to ensure the integrity of updates prior to their use.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TT.UNDETECTED_ ACTIONS | O.AUDIT ensures that activity is monitored so the security of the TOE is not compromised.<br>O.AUTH_COMM ensures that TOE identifies and authenticates all peers prior to communicating with that peer.<br>O.CRYPTOGRAPHY provides the underlying cryptographic functionality required by other protection mechanisms.<br>O.MANAGE requires the TOE to provide mechanisms to allow the TOE be configured in a secure manner.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| TE.ADMIN_FAIL | O.MANAGE provides administrators the capability to update the TOE with security patches.<br>OE.TRUSTED_ADMIN ensures that the administrators are non-hostile and are trained to appropriately manage and administer the TOE.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| P.ACCESS_BANNER | O. DISPLAY_BANNER ensures that the TOE displays banner that provides users with a warning about the unauthorized use of the TOE.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| P.ACCOUNTABILITY | O.ACCESS requires the TOE to identify and authenticate users prior to allowing any TOE access or any TOE mediated access on behalf of those users.<br>O.AUDIT provides the administrator with the capability of reviewing the audit trail based on the identity of the user. Additionally, the administrator’s user identifier is recorded when any security relevant change is made to the TOE (e.g. modifying TSF data, start-stop of the audit mechanism).<br>OE.TRUSTED_ADMIN ensures that the TOE administrators have |

| | |
|---|---|
| | guidance that instructs them how to administer the TOE in a secure manner.<br>O.TIME_STAMPS requires the TOE to provide a reliable time stamp (settable by only the authorized administrator). The audit mechanism is required to include the current date and time in each audit record.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| P.ADMIN_ACCESS | O.MANAGE ensures the TOE to be managed in a secure manner by the authorized administrator.<br>OE.TRUSTED_ADMIN ensures that the administrators of the TOE shall not have any malicious intention, shall receive proper training on the TOE management, and follow the administrator guidelines.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| P.CRYPTOGRAPHIC | O.CRYPTOGRAPHY requires the TOE to implement cryptographic services to provide integrity and confidentiality protection of the TOE.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| A.NO_GENERAL_ PURPOSE | OE.NO_GENERAL_PURPOSE ensures that there are no general-purpose computing capabilities (e.g., the ability to execute arbitrary code or applications) on the TOE.<br>O.THOROUGH_FUNCTIONAL_TESTING ensures that the TOE will undergo appropriate functional testing. |
| A.PHYSICAL | OE.PHYSICAL ensures that the environment provides physical security, commensurate with the value of the TOE and the data it contains. |
| A.TRUSTED_ADMIN | OE.TRUSTED_ADMIN ensures that the administrators of the TOE shall not have any malicious intention, shall receive proper training on the TOE management, and shall follow the administrator guidelines. |

**Table 2: Rationale between Objectives and SPD**

# 5. Extended Components Definition (ASE_ECD)

The following extended component has been included in this Security Target because the Common Criteria components were found to be insufficient as stated.

## 5.1. Extended Component

| Explicit Component | Identifier | Rationale |
|---|---|---|
| FRU_RSA_EXT.1 | Maximum quotas | This extended component is necessary to describe that the TOE specifies the resources in the installation bundle, which get enforced by the virtual infrastructure, and not by the TOE. |

**Table 3: Rationale for Extended Component**

# 6. Security requirements (ASE_REQ)

## 6.1. Security Functional Requirements (SFRs)

| Functional Class | Functional Component | |
|---|---|---|
| FAU: Security audit | FAU_GEN.1 | Audit data generation |
| | FAU_GEN.2 | User identity association |
| FCS: Cryptographic support | FCS_CKM.1 | Cryptographic key generation |
| | FCS_CKM.4 | Cryptographic key destruction |
| | FCS_COP.1 | Cryptographic operation |
| FDP: User data protection | FDP_ACC.1 | Subset access control |
| | FDP_ACF.1 | Security attribute based access control |
| | FDP_IFC.1 | Subset information flow control |
| | FDP_IFF.1 | Simple security attributes |
| FIA: Identification and authentication | FIA_ATD.1 | User attribute definition |
| | FIA_UAU.1 | Timing of authentication |
| | FIA_UID.1 | Timing of identification |
| FMT: Security management | FMT_MSA.1 | Management of security attributes |
| | FMT_MSA.3 | Static attribute initialisation |
| | FMT_MTD.1 | Management of TSF data |
| | FMT_SMF.1 | Specification of Management Functions |
| | FMT_SMR.2 | Restrictions on security roles |
| FPT: Protection of the TSF | FPT_ITT.1 | Basic internal TSF data transfer protection |
| | FPT_PHP.1 | Passive detection of physical attack |
| | FPT_STM.1 | Reliable time stamps |
| | FPT_TST.1 | TSF testing |
| FRU: Resource utilisation | FRU_RSA_EXT.1 | Maximum quotas |
| FTA: TOE access | FTA_SSL.3 | TSF-initiated termination |
| | FTA_TAB.1 | Default TOE access banners |
| FTP: Trusted path/channels | FTP_TRP.1 | Trusted path |

**Table 4: Security Functional Requirements**

## 6.1.1. Security Audit (FAU)

### 6.1.1.1. FAU_GEN.1 Audit data generation

Dependencies: FPT_STM.1 Reliable time stamps

**FAU_GEN.1.1** The TSF shall be able to generate an audit record of the following auditable events:
a) Start-up and shutdown of the audit functions;
b) All auditable events for the [*not specified*] level of audit; and
c) [**All administrative actions, and the following security events:**

- **Failure to save event info,**
- **Failure with service customizing,**
- **Failure to change VM host name,**
- **Failure to create VM network interface,**
- **Failure to reach VM,**
- **Failure to delete network interface in VM,**
- **Failure with ACM client channel**].

**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**].

### 6.1.1.2. FAU_GEN.2 User identity association

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

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

## 6.1.2. Cryptographic Support (FCS)

### 6.1.2.1. FCS_CKM.1 Cryptographic key generation

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

**FCS_CKM.1.1** The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**TLS, SSH and RSA key generation**] and specified cryptographic key sizes [**as specified in Table 5**] that meet the following: [**NIST SP 800-90A, NIST SP 800-135, FIPS PUB 186-4**].

### 6.1.2.2. FCS_CKM.4 Cryptographic key destruction

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.4.1** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**zeroization of keys**] that meets the following: [**FIPS PUB 140-2**].

### 6.1.2.3. FCS_COP.1 Cryptographic operation

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

**FCS_COP.1.1** The TSF shall perform [**cryptographic operations listed in Table 4: Cryptographic Operations**] in accordance with a specified cryptographic algorithm [**listed in Table 4: Cryptographic Operations**] and cryptographic key sizes [**listed in Table 4: Cryptographic Operations**] that meet the following: [**standards listed in Table 4: Cryptographic Operations**].

| Cryptographic operations | Cryptographic algorithm | Key sizes (bits) | Standards |
|---|---|---|---|
| Encryption/decryption | AES | 128,192,256 | FIPS PUB 197 |
| Encryption/decryption | RSA | 2048 | NA |
| Encryption/decryption | 3DES | 168 | FIPS PUB 46-3 |
| TLS Session Keys Generation | TLS KDF | All TLS Session Key Sizes | NIST SP 800-90A<br>NIST SP 800-135 |
| SSH Session Key Generation | SSH KDF | All SSH Session Key Sizes | NIST SP 800-90A<br>NIST SP 800-135 |
| Key Generation | RSA | 2048 | FIPS PUB 186-4 |
| Digital Signature | ECDSA | 256,384,521 | FIPS PUB 186-4 |
| Digital Signature | RSA | 2048 | FIPS PUB 186-4 |
| Digital Signature | DSA | 2048 | FIPS PUB 186-4 |
| Hashing | SHA-1,<br>SHA-256,<br>SHA-384,<br>SHA-512 | | FIPS PUB 180-4 |
| HMAC | HMAC-SHA-1,<br>HMAC-SHA-256,<br>HMAC-SHA-384,<br>HMAC-SHA-512 | 160,256,384,512 | FIPS PUB 198-1 |
| Key Agreement | DH | 2048 | NA |
| Key Agreement | ECDH | 256,384,521 | NA |

**Table 5: Cryptographic Operations**

## 6.1.3. User Data Protection (FDP)

### 6.1.3.1. FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

**FDP_ACC.1.1** The TSF shall enforce the [**BOTsink Access Control Policy**] on [subjects: **TOE users**, objects: **TOE data**, operations: **access to TOE data**].

### 6.1.3.2. FDP_ACF.1 Security attribute based access control

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

**FDP_ACF.1.1** The TSF shall enforce the [**BOTsink Access Control Policy**] to objects based on the following: [subjects: **TOE users**; subject attributes: **user name, password**; object: **TOE data**; object attributes: **none**].

**FDP_ACF.1.2** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [**no additional rules**].

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

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

### 6.1.3.3. FDP_IFC.1 Subset information flow control

Dependencies: FDP_IFF.1 Simple security attributes

**FDP_IFC.1.1** The TSF shall enforce the [**Information Flow Control SFP**] on [

- subjects: **external IT entities that send information towards the TOE**,
- information: **traffic sent towards the TOE**,
- operation: **allow/reject information**].

### 6.1.3.4. FDP_IFF.1 Simple security attributes

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

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

- subjects: **external IT entities that send information towards the TOE**,
- subject security attributes: **address**,
- information: **traffic sent towards the TOE**,
- information security attributes: **destination address and port**].

**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: [**subjects on the deployment network can cause information to flow towards the TOE if the information security attribute values are permitted by the information flow security policy rules**].

**FDP_IFF.1.3** The TSF shall enforce the [**no additional information flow control SFP rules**].

**FDP_IFF.1.4** The TSF shall explicitly authorise an information flow based on the following rules: [**no additional information flow control SFP rules**].

**FDP_IFF.1.5** The TSF shall explicitly deny an information flow based on the following rules: [**reject requests for access or service if the information security attribute values are not permitted by the information flow security policy rules**].

## 6.1.4. Identification and Authentication (FIA)

### 6.1.4.1. FIA_ATD.1 User attribute definition

Dependences: None.

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

a) User identity: **user name;**
b) Local authentication data: **password;**
c) Authorizations: **privileges;** and
d) **first name, last name**].

### 6.1.4.2. FIA_UAU.1 Timing of authentication

Dependences: FIA_UID.1 Timing of identification

**FIA_UAU.1.1** The TSF shall allow [**entry of user name and corresponding password**] on behalf of the user to be performed before the user is authenticated.

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

### 6.1.4.3. FIA_UID.1 Timing of identification

Dependences: None.

**FIA_UID.1.1** The TSF shall allow [**entry of user name and corresponding password**] on behalf of the user to be performed before the user is identified.

**FIA_UID.1.2** The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

## 6.1.5. Security management (FMT)

### 6.1.5.1. FMT_MSA.1 Management of security attributes

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

**FMT_MSA.1.1** The TSF shall enforce the [**Information Flow Control SFP**] to restrict the ability to [ [***manage***]] the security attributes [**as in FDP_IFF.1.1**] to [**authorized administrator**].

### 6.1.5.2. FMT_MSA.3 Static attribute initialisation

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

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

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

#### 6.1.5.3. FMT_MTD.1 Management of TSF data

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

**FMT_MTD.1.1** The TSF shall restrict the ability to [***manage***] the [**TSF data**] to [**authorized administrator**].

#### 6.1.5.4. FMT_SMF.1 Specification of Management Functions

Dependencies: None.

**FMT_SMF.1.1** The TSF shall be capable of performing the following management functions: [**administer the TOE locally and remotely**].

#### 6.1.5.5. FMT_SMR.2 Restrictions on security roles

Dependencies: FIA_UID.1 Timing of identification

**FMT_SMR.2.1** The TSF shall maintain the roles: [**authorized command line administrator, authorized web-based administrator, authorized web-based user**].

**FMT_SMR.2.2** The TSF shall be able to associate users with roles.

**FMT_SMR.2.3** The TSF shall ensure that the conditions [**authorized administrator role shall administer the TOE locally and remotely**] are satisfied.

### 6.1.6. Protection of the TSF (FPT)

#### 6.1.6.1. FPT_ITT.1 Basic internal TSF data transfer protection

Dependencies: None.

**FPT_ITT.1.1** The TSF shall protect TSF data from [*disclosure, modification*] when it is transmitted between separate parts of the TOE.

#### 6.1.6.2. FPT_PHP.1 Passive detection of physical attack

Dependencies: None.

**FPT_PHP.1.1** The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

**FPT_PHP.1.2** The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

#### 6.1.6.3. FPT_STM.1 Reliable time stamps

Dependencies: None.

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

#### 6.1.6.4. FPT_TST.1 TSF testing

Dependencies: None.

**FPT_TST.1.1** The TSF shall run a suite of self tests [*at the conditions* ***where the authorised user can***

- ***enable the test traffic feature, and***
- ***specify the frequency for sending the test traffic***] to demonstrate the correct operation of [*the TSF*].

**FPT_TST.1.2** The TSF shall provide authorised users with the capability to verify the integrity of [ *No TSF data*].

**FPT_TST.1.3** The TSF shall provide authorised users with the capability to verify the integrity of [ *No parts of TSF*].

## 6.1.7. Resource utilisation (FRU)

### 6.1.7.1. FRU_RSA_EXT.1 Maximum quotas

Dependencies: None.

**FRU_RSA_EXT.1.1** The TSF shall specify maximum quotas of the number of the virtual CPUs in the installation bundle, enforced by the virtual infrastructure.

## 6.1.8. TOE access (FTA)

### 6.1.8.1. FTA_SSL.3 TSF-initiated termination

Dependencies: None.

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [**fixed timeout period of 5 minutes or a configurable timeout period of 10-60 minutes**].

### 6.1.8.2. FTA_TAB.1 Default TOE access banners

Dependencies: None.

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

## 6.1.9. Trusted path/channels (FTP)

### 6.1.9.1. FTP_TRP.1 Trusted path

Dependencies: None.

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

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

**FTP_TRP.1.3** The TSF shall require the use of the trusted path for [*initial user authentication,* [***all remote administration actions***]].

## 6.2. Security assurance requirements (SARs)

The assurance level of the TOE is EAL2 augmented with ALC_FLR.1.

| Assurance Class | Assurance Components |
|---|---|

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

**Table 6: Assurance requirements**

## 6.3. Security Requirements Rationale

### 6.3.1. Relation between security requirements and security objectives

The following tracing shows which SFRs address which security objectives for the TOE, and it shows which TOE security objectives are adressed by SARs.

| Requirements<br>Objectives | FAU_GEN.1 | FAU_GEN.2 | FCS_CKM.1 | FCS_CKM.4 | FCS_COP.1 | FDP_ACC.1 | FDP_ACF.1 | FDP_IFC.1 | FDP_IFF.1 | FIA_ATD.1 | FIA_UAU.1 | FIA_UID.1 | FMT_MSA.1 | FMT_MSA.3 | FMT_MTD.1 | FMT_SMF.1 | FMT_SMR.2 | FPT_ITT.1 | FPT_PHP.1 | FPT_STM.1 | FPT_TST.1 | FRU_RSA_EXT.1 | FTA_SSL.3 | FTA_TAB.1 | FTP_TRP.1 | SARs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| O.ACCESS | | | | | | | | | | X | X | X | | | | | | | | | | | X | | X | |
| O.AUDIT | X | X | | | | | | | | | | | | | | | | | | X | | | | | | |
| O.AUTH_COMM | | | X | | X | | | | | | | | | | | | | | | | | | | | | |
| O.CHANGE_ MANAGEMENT | | | | | | | | | | | | | | | | | | | | | | | | | | X |
| O.CRYPTO-GRAPHY | | | X | X | X | | | | | | | | | | | | | | | | | | | | | |
| O.DISPLAY_ BANNER | | | | | | | | | | | | | | | | | | | | | | | | X | | |
| O.MANAGE | | | | | | | | | | | | | X | X | X | X | X | | | | X | | | | | |
| O.MEDIATE | | | | | | X | X | | | | | | | | | | | | | | | | | | | |
| O.NETWORK_ FILTERING | | | | | | | | X | X | | | | | | | | | | | | | | | | | |
| O.REPLAY_ DETECTION | | | | | | | | | | X | | | | | | | | | | | | | | | | |
| O.RESOURCE | | | | | | | | | | | | | | | | | | | | | | X | | | | |
| O.SELF_ PROTECTION | | | | | | | | | | | | | | | | | | X | X | | | | | | | |
| O.SOUND_ | | | | | | | | | | | | | | | | | | | | | | | | | | X |

| Requirements | FAU_GEN.1 | FAU_GEN.2 | FCS_CKM.1 | FCS_CKM.4 | FCS_COP.1 | FDP_ACC.1 | FDP_ACF.1 | FDP_IFC.1 | FDP_IFF.1 | FIA_ATD.1 | FIA_UAU.1 | FIA_UID.1 | FMT_MSA.1 | FMT_MSA.3 | FMT_MTD.1 | FMT_SMF.1 | FMT_SMR.2 | FPT_ITT.1 | FPT_PHP.1 | FPT_STM.1 | FPT_TST.1 | FRU_RSA_EXT.1 | FTA_SSL.3 | FTA_TAB.1 | FTP_TRP.1 | SARs |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Objectives | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DESIGN | | | | | | | | | | | | | | | | | | | | | | | | | | |
| O.TIME_ STAMPS | | | | | | | | | | | | | | | X | | | | | X | | | | | | |
| O.THOROUGH_ FUNCTIONAL_ TESTING | | | | | | | | | | | | | | | | | | | | | | | | | | X |
| O.VERIFIABLE_ UPDATES | | | | | X | | | | | | | | | | | | | | | | | | | | | |

**Table 7: Tracing of functional requirements to Objectives**

The following set of justifications shows that all security objectives for the TOE are effectively addressed by the SFRs.

| Security Objectives | Security Functional Requirement Rationale |
|---|---|
| O.ACCESS | FIA_ATD.1 defines the attributes of users, including a user identifier that is used to by the TOE to determine a user's identity and enforce what type of access the user has to the TOE, and ensures that untrusted users cannot be associated with a role and reduces the possibility of a user obtaining administrative privileges.<br>FIA_UAU.1 ensures that users are authenticated before they are provided access to the TOE or its services. In order to control logical access to the TOE an authentication mechanism is required. The local user authentication mechanism is necessary to ensure that an administrator has the ability to login to the TOE regardless of network connectivity (e.g., it would be unacceptable if an administrator could not login to the TOE because the authentication server was down, or that the network path to the authentication server was unavailable).<br>FIA_UID.1 ensures that every user is identified before the TOE performs any mediated functions.<br>FTA_SSL.3 ensures that inactive user sessions are dropped.<br>FTP_TRP.1 ensures that remote users have a trusted path in order to authenticate. |
| O.AUDIT | FAU_GEN.1 defines the set of events that the TOE must be capable of recording. This requirement ensures that the administrator can audit security relevant events that take place in the TOE.<br>FAU_GEN.2 ensures that the audit records associate a user identity with the auditable event. In the case of authorized users, the association is accomplished with the user ID.<br>FPT_STM.1 supports the audit functionality by ensuring that the TOE is capable of obtaining a time stamp for use in recording audit events. |
| O.AUTH_COMM | FCS_CKM.1 and FCS_COP.1 provide a mechanism that creates a distinct communication channel between the TOE and both remote administrators and trusted IT entities that |

| | |
|---|---|
| | protects the data that traverse this channel from disclosure or modification. This is done cryptographically using the protocols specified by the requirements; these protocols provide the assured mutual identification of the endpoints and protection of the channel data. |
| O.CHANGE_ MANAGEMENT | ALC_CMC.2 requires the developer to use a CM system that uniquely identifies all configuration items.<br>ALC_CMS.2 defines the configuration list, which must include the TOE, the parts that comprise the TOE, and the evaluation evidence. These configuration items are controlled in accordance with CM capabilities (ALC_CMC.2).<br>ALC_DEL.1 states that the delivery documentation shall describe all procedures used to maintain security of the TOE when distributing the TOE to the user.<br>ALC_FLR.1 requires the developer to establish flaw remediation procedures that describe the tracking of security flaws, the identification of corrective actions, and the distribution of corrective action information to TOE users. |
| O.CRYPTOGRAPHY | FCS_CKM.1 ensures that the TOE is capable of generating cryptographic keys.<br>FCS_CKM.4 provides the functionality for ensuring key and key material is zeroized.<br>FCS_COP.1 requires that for data decryption and encryption FIPS PUB 140-2 standard approved algorithms are used. |
| O.DISPLAY_BANNER | FTA_TAB.1 ensures that a user will have to have some type of display device for TOE usage, and therefore a notice and consent banner is required. |
| O.MANAGE | The FMT requirements are used to satisfy this management objective, as well as other objectives that specify the control of functionality. The requirements' rationale for this objective focuses on the administrator's capability to perform management functions in order to control the behavior of security functions.<br>FMT_MSA.1, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1 and FMT_SMR.2 ensure that only the authorized administrator role can manage the TOE, and that the TOE supports both local and remote administration.<br>FPT_TST.1 provides capability for the authorised administrator to enable and configure the test traffic feature. |
| O.MEDIATE | FDP_ACC.1 defines the Access Control policy that will be enforced on subjects acting on behalf of users attempting to gain access to objects. All operations that involve access to the data are controlled by the policy. These objects contain the TOE data to be protected.<br>FDP_ACF.1 defines the security attributes used to provide access control to objects based on the TOE's access control policy. |
| O.NETWORK_ FILTERING | FDP_IFC.1 identifies the external IT entities in the Information Flow Control SFP that send information towards the TOE. The SFP will either reject or allow the information flow. |

| | |
|---|---|
| | FDP_IFF.1 identifies the external IT entity and its security attributes as part of the information flow control SFP. TOE will permit or deny the information flow based on network packet rules managed by administrator. |
| O.REPLAY_ DETECTION | FIA_ATD.1 provides users with attributes to distinguish one user from another, for accountability purposes, and to associate privileges with users. |
| O.RESOURCE | FRU_RSA_EXT.1 provides the capability to mitigate attempts to exhaust the resources provided by the TOE. |
| O.SELF_PROTECTION | FPT_ITT.1 ensures that TSF data that is transmitted between components of the TOE is protected against disclosure and tampering. This would also include any TSF data that is sent from an administrative console to the TOE if that console is "networked" with the TOE. This would not apply to TSF data that is configured from a console that is connected via a communication path (e.g., serial cable) that ensures the data cannot be disclosed. The disclosure of TSF data could create an opportunity for the TOE to be rendered ineffective in enforcing its security policies.<br>FPT_PHP.1 provides detection capability of TOE when physical tampering with the TOE occurs. |
| O.SOUND_DESIGN | ADV_ARC.1 requires the TSF to be structured such that it cannot be tampered with or bypassed, and that TSFs provide security domains that isolate those domains from each other.<br>ADV_FSP.2 requires the developer to provide a description of the TSF interfaces in terms of their purpose, method of use, and parameters. In addition, the security relevant actions, results and error messages of each TSF interface that is security relevant shall also be described.<br>ADV_TDS.1 requires that the architecture of the TOE shall be described in terms of subsystems. It identifies which subsystems are responsible for making and enforcing security relevant decisions, and provides a description of how those decisions are made and enforced. |
| O.TIME_STAMPS | FMT_MTD.1 provides the capability to set the time used for generating time stamps to the authorized administrator. This functionality allows the authorized administrator to ensure the time and date are correctly set, while restricting this function from unauthorized use.<br>FPT_STM.1 requires that the TOE be able to provide reliable time stamps for its own use. Time stamps include date and time and are reliable in that they are always available to the TOE. |
| O.THOROUGH_ FUNCTIONAL_ TESTING | ATE_FUN.1 requires the developer to correctly perform and document the tests in the test documentation.<br>ATE_IND.2 determines whether the TOE behaves as specified in the design documentation by independently testing a subset of the TSF, , and gains confidence in the developer's test results by performing a sample of the developer's tests. |
| O.VERIFIABLE_UPD ATES | FCS_COP.1 ensures that the update was downloaded via secure communications. |

**Table 8: Rationale between Objectives and SFRs and SARs**

## 6.3.2. SFR Dependencies

The table below shows the dependencies of the security functional requirements of the TOE and gives a rationale for each of them if they are included or not.

| Security functional requirement | Dependency | Dependency Rationale |
|---|---|---|
| FAU_GEN.1 Audit data generation | FPT_STM.1 Reliable time stamps | Included |
| FAU_GEN.2 User identity association | FAU_GEN.1 Audit data generation<br>FIA_UID.1 Timing of identification | Included |
| FCS_CKM.1 Cryptographic key generation | [FCS_CKM.2 Cryptographic key distribution, or<br>FCS_COP.1 Cryptographic operation]<br>FCS_CKM.4 Cryptographic key destruction | Included |
| FCS_CKM.4 Cryptographic key destruction | [FDP_ITC.1 Import of user data without security attributes, or<br>FDP_ITC.2 Import of user data with security attributes, or<br>FCS_CKM.1 Cryptographic key generation] | Included |
| FCS_COP.1 Cryptographic Operation | [FDP_ITC.1 Import of user data without security attributes, or<br>FDP_ITC.2 Import of user data with security attributes, or<br>FCS_CKM.1 Cryptographic key generation]<br>FCS_CKM.4 Cryptographic key destruction | Included |
| FDP_ACC.1 Subset access control | FDP_ACF.1 Security attribute based access control | Included |
| FDP_ACF.1 Security attribute based access control | FDP_ACC.1 Subset access control<br>FMT_MSA.3 Static attribute initialisation | Included |
| FDP_IFC.1 Subset information flow control | FDP_IFF.1 Simple security attributes | Included |
| FDP_IFF.1 Simple security attributes | FDP_IFC.1 Subset information flow control<br>FMT_MSA.3 Static attribute initialisation | Included |
| FIA_ATD.1 User attribute definition | None | |
| FIA_UAU.1 Timing of authentication | FIA_UID.1 Timing of identification | Included |
| FIA_UID.1 Timing of identification | None | |

| Security functional requirement | Dependency | Dependency Rationale |
|---|---|---|
| FMT_MSA.1 Management of security attributes | [FDP_ACC.1 Subset access control, or<br>FDP_IFC.1 Subset information flow control]<br>FMT_SMR.1 Security roles<br>FMT_SMF.1 Specification of Management Functions | Included[1] |
| FMT_MSA.3 Static attribute initialisation | FMT_MSA.1 Management of security attributes<br>FMT_SMR.1 Security roles | Included[2] |
| FMT_MTD.1 Management of TSF data | FMT_SMR.1 Security roles<br>FMT_SMF.1 Specification of Management Functions | Included[3] |
| FMT_SMF.1 Specification of Management Functions | None | |
| FMT_SMR.2 Restrictions on security roles | FIA_UID.1 Timing of identification | Included |
| FPT_ITT.1 Basic internal TSF data transfer protection | None | |
| FPT_PHP.1 Passive detection of physical attack | None | |
| FPT_STM.1 Reliable time stamps | None | |
| FPT_TST.1 TSF testing | None | |
| FRU_RSA_EXT.1 Maximum Quotas | None | |
| FTA_SSL.3 TSF-initiated termination | None | |
| FTA_TAB.1 Default TOE access banners | None | |
| FTP_TRP.1 Trusted path | None | |

**Table 9: SFR's dependencies and rationale**

### 6.3.3. SAR Rationale

The SARs specified in this ST are according to EAL2, augmented with ALC_FLR.1.

[1] FMT_MSA.1 has a dependency to FMT_SMR.1 which is covered by FMT_SMR.2.

[2] FMT_MSA.3 has a dependency to FMT_SMR.1 which is covered by FMT_SMR.2.

[3] FMT_MTD.1 has a dependency to FMT_SMR.1 which is covered by FMT_SMR.2.

# 7. TOE Summary Specification (ASE_TSS)

## 7.1. TOE Security Functions Specification

This section describes the security functions provided by the TOE to meet the security functional requirements specified for the TOE in section 6.1 Security Functional Requirements (SFRs).

The table below shows the mapping between the SFRs and the implementing security functions, and a description is given in the following subsections.

| Requirements<br>Functions | FAU_GEN.1 | FAU_GEN.2 | FCS_CKM.1 | FCS_CKM.4 | FCS_COP.1 | FDP_ACC.1 | FDP_ACF.1 | FDP_IFC.1 | FDP_IFF.1 | FIA_ATD.1 | FIA_UAU.1 | FIA_UID.1 | FMT_MSA.1 | FMT_MSA.3 | FMT_MTD.1 | FMT_SMF.1 | FMT_SMR.2 | FPT_ITT.1 | FPT_PHP.1 | FPT_STM.1 | FPT_TST.1 | FRU_RSA_EXT.1 | FTA_SSL.3 | FTA_TAB.1 | FTP_TRP.1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SF.TOE_ACCESS_ FUNCTIONS | | | | | | X | X | | | X | X | X | | | | | | | | | | | X | X | X |
| SF.SECURITY_ AUDIT | X | X | | | | | | | | | | | | | | | | | | X | | | | | |
| SF.CRYPTO-GRAPHIC_ SUPPORT | | | X | X | X | | | | | | | | | | | | | | | | | | | | |
| SF.SECURITY_ MANAGEMENT | | | | | | | | | | | | | X | X | X | X | X | | | | X | | | | |
| SF.COMMUNI-CATION | | | | | | | | X | X | | | | | | | | | | | | | X | | | |
| SF.SECURITY_ PROTECTION | | | | | | | | | | | | | | | | | | X | X | | | | | | |

**Table 10: Mapping SFRs to security functions**

### 7.1.1. SF. TOE_ACCESS_FUNCTIONS

**(FDP_ACC.1, FDP_ACF.1)**
The TOE provides the user data protection security function requirement to manage user access and interaction with TOE data. The TOE enforces access control policy which limits access to the TOE data. Access to data is enforced by user account privileges (permissions). A user attempting to access the TOE with the incorrect privileges will be denied access.
Users can access the TOE through a GUI interface, a secure CLI administrator interface on the device console, the serial console or SSH interface. Access to the TOE through these interfaces requires the correct access type associated with each TOE user. Once granted access to the TOE, the users with the correct privileges can manage TOE data. TOE users can be denied access to the TOE, if they are attempting to access the TOE from an interface of incorrect access type.

**(FIA_ATD.1)**
User account information is stored in the TOE and contains the following attributes for users:

- Privileges – Select 'Admin' to provide admin privileges to the user. Admin privileges are required to access the configuration of the TOE.
- User name – The logon name (case-sensitive) of the user. For AD users, logon name without the domain name must be used. If it is selected in AD that the user must change password at next logon, and if the user is using the AD logon credential for the first time, logging on fails.
- First name – The user's first name.
- Last name – The user's second name.

- Password – The user must use a strong password. This field is available for local users only.

**(FIA_UAU.1, FIA_UID.1)**
Each individual must be successfully identified and authenticated with a username and password by the TSF before access is allowed to the TOE. User identification and authentication by the TSF uses the security attributes of the user account described above.
When identification and authentication data is entered, the TOE verifies if the user name is to be locally authenticated or authenticated by the Windows Active Directory.
If the user name is to be locally authenticated, the TOE attempts to identify an applicable local user account from the provided identity and if a match is found, the password provided is checked against the user account information in the internal database. If a user account cannot be associated with the provided identity or the provided password does not match the user account information, identification and authentication will fail.
If the user name is to be authenticated by the Windows Active Directory, the authentication access is granted if the authentication succeeds against the AD server.
No actions are allowed, other than entry of identification and authentication data, until identification and authentication are successful.

**(FTA_SSL.3)**
Both local and remote user sessions have defined maximum inactivity times. When a session is inactive (i.e., no session input) for the configured period of time between 10 and 60 minutes or a fixed period of 5 minutes, the TOE will terminate the session, requiring the administrator to log in again to establish a new session when needed.

**(FTA_TAB.1)**
The TOE has the functionality to present warning information to a user when attempting to login. The banners for the CLI and the GUI are configurable through Web-based management interface.

**(FTP_TRP.1)**
The TOE uses SSH or HTTPS to provide the trusted path (with protection from disclosure and modification) for all remote user sessions.

### 7.1.2. SF. SECURITY_AUDIT

**(FAU_GEN.1)**
The TOE generates a comprehensive set of audit logs that identify specific TOE operations whenever an auditable event occurs. Shutdown and start-up of the audit functions are logged by events for reloading the TOE, and the events when the TOE comes back up. The possible severity levels are – INFO, ERROR, and WARNING.

**(FAU_GEN.2)**
The TOE ensures that each auditable event is associated with the user that triggered the event. For example, user identity or related session ID would be included in the audit record for a human user.

**(FPT_STM.1)**
The TOE provides a source of date and time information used in audit event timestamps. The default time zone of the xACMs and the xBOTsinks is UTC. The TOE can optionally be set to receive clock updates from an NTP server. This date and time is used as the timestamp that is applied to TOE generated audit record(s).

### 7.1.3. SF.CRYPTOGRAPHIC_SUPPORT

The vBOTsink uses FIPS 140-2 Security Level 1 (software only) compliant cryptographic modules. The BOTsink and ACM are compliant to FIPS 140-2 Security Level 2 (includes hardware).

**(FCS_CKM.1)**
In support of secure cryptographic protocols, the TOE supports several key generation schemes, including RSA as specified in FIPS PUB 186-4 and schemes used by TLS and SSH as specified in NIST SP 800-90A and NIST SP 800-135.

**(FCS_CKM.4)**
The TOE meets all requirements specified in FIPS 140-2 for destruction of keys. All keys within the TOE are zeroizable.

**(FCS_COP.1)**
The TOE provides encryption and decryption capabilities

- using 128, 192 and 256 bits AES, described in FIPS PUB 197,
- using 2048 bit RSA,
- using 168 bit 3DES, described in FIPS PUB 46-3.

The TOE provides key generation capabilities

- using TLS, as described in NIST SP 800-90A and NIST SP 800-135,
- using SSH, as described in NIST SP 800-90A and NIST SP 800-135,
- using 2048 bit RSA, as described in FIPS PUB 186-4.

The TOE provides digital signature capabilities

- using 256, 384 and 521 bits ECDSA, as described in FIPS PUB 186-4,
- using 2048 bit RSA, as described in FIPS PUB 186-4,
- using 2048 DSA, as described in FIPS PUB 186-4.

The TOE provides Hashing capabilities using SHA-1, SHA-256, SHA-384, and SHA-512, described in FIPS PUB 180-4.
The TOE provides HMAC capabilities using 160, 256, 384 and 512 bits HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512, described in FIPS PUB 198-1.
The TOE provides Key Agreement capabilities using 2048 DH.
The TOE provides capabilities using 256, 384 and 521 bits ECDH.

### 7.1.4. SF.SECURITY_MANAGEMENT

**(FMT_MSA.1, FMT_MSA.3)**
The TOE ensures that only authorized administrators are able to specify the policy definitions to enforce availability. By default, the TOE provides a restrictive information flow policy ruleset.

**(FMT_MTD.1)**
The TOE provides the ability for authorized administrators to access TOE data, such as audit data, configuration data, security attributes, session thresholds and updates. For web-based management users, each of the predefined privilege levels has a set of permissions that grants access to the TOE data, though with some privilege levels the access is limited. For the purposes of this evaluation, the ‘admin’ privilege level refers to authorized CLI administrator privilege level or authorized GUI administrator privilege level.

**(FMT_SMF.1)**
The TOE provides all the capabilities necessary to securely manage the TOE. The administrative user can connect to the TOE either via CLI or through the TOE GUI to perform these functions. However, the specific configurable parameters available through the TOE CLI are limited. All general administration is expected to take place through the TOE GUI. The specific management capabilities available from the TOE include:

- Local and remote administration of the TOE services and security characteristics;
- The ability to update the TOE software;
- Ability to configure the SSH functionality;
- Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE via the GUI.

**(FMT_SMR.2)**
The term "authorized administrator" is used in this ST to refer to any user which has been assigned to an 'admin' privilege level that is permitted to perform the relevant action and therefore has the appropriate privileges to perform the requested functions. The assigned security role determines the functions the user can perform. The TOE authenticates all access to the administrative interfaces using a username and password. The TOE supports both local and remote administration.

**(FPT_TST.1)**
The TOE provides the ability for authorized administrators to configure test traffic to be sent towards itself at certain time intervals. This enables verification of TSF function such as administrative alerts.

### 7.1.5. SF.COMMUNICATION

**(FDP_IFC.1, FDP_IFF.1)**
The TOE is protected from deployment network intrusions by using information flow control. The TOE will check the IP packets from external IT entities before allowing or rejecting the network traffic in forms of IP packets. The decision to allow or reject/drop traffic will be based on packet filter rules managed by administrator.
The TOE could configure packet filter rules and policies based on the subject and information security attributes or criteria. The criteria are as follows:

- Source address of information (i.e IP address, MAC address)
- Destination address of information (i.e IP address, MAC address)
- Source port of information
- Destination port of information

**(FRU_RSA_EXT.1)**
The TOE shall optimize security related communication by requiring the TOE to provide controls relating to CPU resources, implemented by the virtual infrastructure, preventing the communication of occupying all CPU resources.

### 7.1.6. SF.SECURITY_PROTECTION

**(FPT_ITT.1)**
The TSF ensures that data transmitted between separate parts of the TOE are protected from disclosure or modification. This protection is ensured by transmission of data over secure TLS channels between endpoint agents and xBOTsinks, and between ACMs and xBOTsinks.

**(FPT_PHP.1)**
The TOE detects physical tampering. The BOTsink and ACM physical appliances are FIPS 140-2 Level 2 compliant.