AhnLab MDS, MDS with MTA, and MDS Manager V2.1

Security Target

Version 0.92
December 20, 2018

Prepared for:

AhnLab

673 Sampyeong-dong,
Bundang-gu, Seongnam-si, Gyeonggi-do, 463-400
Korea

Prepared by:

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive, Columbia, Maryland 21046

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is a single instance of AhnLab MDS, MDS with MTA, and MDS Manager V2.1. In this ST, the TOE instances may be collectively referred to as the AhnLab MDS.

AhnLab Malware Defense System (MDS) and MDS with MTA are network devices for an Autonomous Malware Defense System (MDS) that combines signature, signature-less and reputation feed methods together, to detect both known and unknown malware entering the network. MDS with MTA is a network device that includes MTA functionality via purchase of separate license. The MDS Manager Device is a network device that can be used to manage other MDS devices, though this capability is not covered in the evaluation. Each device supports the security functions specified in [NDcPP].

The focus of this evaluation is on the TOE functionality supporting the claims in the Collaborative Protection Profile for Network Devices (NDcPP) (See section 1.2 for specific version information). The only capabilities covered by the evaluation are those specified in the [NDcPP] Protection Profile, all other capabilities are not covered in the evaluation. The security functionality specified in [NDcPP] includes identification and authentication of administrators, auditing of security-relevant events, ability to verify the source and integrity of updates to the TOE, protected communications for administration and TOE operation, and specifies CAVP-validated cryptographic mechanisms.

The Security Target contains the following additional sections:

- TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – AhnLab MDS, MDS with MTA, and MDS Manager V2.1 Security Target

ST Version – Version 0.92

ST Date – December 20, 2018

TOE Identification – AhnLab MDS, MDS with MTA, and MDS Manager V2.1

TOE Versions:

Product Series	Specific Product Device	Device Software
MDS	MDS 1000	MDS Analyzer: 2.1.8.29 Data Viewer: 2.1.8.31 Host Controller: 2.1.7.22
	MDS 2000	
	MDS 4000	
	MDS 4000A	
	MDS 6000	
	MDS 8000	
	MDS 8000A	
	MDS10000	
	MDS10000A	
MDS with MTA	MDS 4000	MDS Analyzer: 2.1.8.29 Data Viewer: 2.1.8.31 Host Controller: 2.1.7.22
	MDS 4000A	
	MDS 6000	
	MDS 8000	
	MDS 8000A	
	MDS 10000	
	MDS10000A	
MDS Manager	MDS Manager 2000	Data Viewer: 2.1.8.31 Host Controller: 2.1.7.22
	MDS Manager 5000R	Data Viewer: 2.1.8.31 Host Controller: 2.1.7.22
	MDS Manager 5000AR	Data Viewer: 2.1.8.31 Host Controller: 2.1.7.22
	MDS Manager 10000R	Data Viewer: 2.1.8.31 Host Controller: 2.1.7.22
	MDS Manager 10000AR	Data Viewer: 2.1.8.31 Host Controller: 2.1.7.22

TOE Developer – AhnLab

Evaluation Sponsor – AhnLab

CC Identification – *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012*

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- *collaborative Protection Profile for Network Devices Version 1.0, 27 February 2015,* [CPPND] and including the following optional SFRs: FAU_STG.1, FCS_HTTPS_EXT.1, FCS_SSHS_EXT.1, FCS_TLSC_EXT.2, and FCS_TLSS_EXT.1. The following NIAP Technical Decisions apply to this PP and have been accounted for in the ST development and the conduct of the evaluation:

- TD0090: NIT Technical Decision for FMT_SMF.1.1 Requirement in NDcPP
- TD0094: NIT Technical Decision for validating a published hash in NDcPP
- TD0095: NIT Technical Interpretations regarding audit, random bit generation, and entropy in NDcPP
- TD0111: NIT Technical Decision for third party libraries and FCS_CKM.1 in NDcPP and FWcPP
- TD0112: NIT Technical Decision for TLS testing in the NDcPP v1.0 and FW cPP v1.0.
- TD0113: NIT Technical Decision for testing and trusted updates in the NDcPP v1.0 and FW cPP v1.0
- TD0114: NIT Technical Decision for Re-Use of FIPS test results in NDcPP and FWcPP
- TD0115: NIT Technical Decision for Transport mode and tunnel mode in IPsec communication in NDcPP and FWcPP
- TD0116: NIT Technical Decision for a Typo in reference to RSASSA-PKCS1v1_5 in NDcPP and FWcPP
- TD0117 (supercedes TD0093): NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP
- TD0125: NIT Technical Decision for Checking validity of peer certificates for HTTPS servers
- TD0126: NIT Technical Decision for TLS Mutual Authentication
- TD0130: NIT Technical Decision for Requirements for Destruction of Cryptographic Keys
- TD0143: NIT Technical Decision for Failure testing for TLS session establishment in NDcPP and FWcPP
- TD0150: NIT Technical Decision for Removal of SSH re-key audit events in the NDcPP v1.0 and FW cPP v1.0
- TD0151: NIT Technical Decision for FCS_TLSS_EXT Testing - Issue 1 in NDcPP v1.0.
- TD0152: NIT Technical Decision for Reference identifiers for TLS in the NDcPP v1.0 and FW cPP v1.0
- TD0153: NIT Technical Decision for Auditing of NTP Time Changes in the NDcPP v1.0 and FW cPP v1.0
- TD0154: NIT Technical Decision for Versions of TOE Software in the NDcPP v1.0 and FW cPP v1.0
- TD0155: NIT Technical Decision for TLSS tests using ECDHE in the NDcPP v1.0.
- TD0156: NIT Technical Decision for SSL/TLS Version Testing in the NDcPP v1.0 and FW cPP v1.0
- TD0160: NIT Technical Decision for Transport mode and tunnel mode in IPSEC communications
- TD0164: NIT Technical Decision for Negative testing for additional ciphers for SSH
- TD0165: NIT Technical Decision for Sending the ServerKeyExchange message when using RSA
- TD0167: NIT Technical Decision for Testing SSH 2^28 packets
- TD0168: NIT Technical Decision for Mandatory requirement for CSR generation

- TD0169: NIT Technical Decision for Compliance to RFC5759 and RFC5280 for using CRLs
- TD0170: NIT Technical Decision for SNMPv3 Support
- TD0181: NIT Technical Decision for Self-testing of integrity of firmware and software
- TD0182: NIT Technical Decision for Handling of X.509 certificates related to ssh-rsa and remote comms
- TD0183: NIT Technical Decision for Use of the Supporting Document
- TD0184: NIT Technical Decision for Mandatory use of X.509 certificates
- TD0185: NIT Technical Decision for Channel for Secure Update
- TD0186: NIT Technical Decision for Applicability of X.509 certificate testing to IPsec
- TD0187: NIT Technical Decision for Clarifying FIA_X509_EXT.1 test 1
- TD0188: NIT Technical Decision for Optional use of X.509 certificates for digital signatures
- TD0189: NIT Technical Decision for SSH Server Encryption Algorithms
- TD0191: NIT Technical Decision for Using secp521r1 for TLS communication
- TD0195: NIT Technical Decision Making DH Group 14 optional in FCS_IPSEC_EXT.1.11[1]

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012
  - Part 3 Conformant

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a number in parentheses placed at the end of the component. For example, FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, (1) and (2).
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that

[1] The TOE does not claim IPsec, therefore TD0195 is not applicable.

'cases' that are not applicable in a given SFR have simply been removed without any explicit identification.

- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.1 Abbreviations and Acronyms

ACL	Access Control List
AES	Advanced Encryption Standard
APT	Advanced Persistent Threat
CBC	Cipher-Block Chaining
CC	Common Criteria for Information Technology Security Evaluation
CLI	Command Line Interface
DH	Diffie-Hellman
FIPS	Federal Information Processing Standard
HMAC	Hashed Message Authentication Code
HTTPS	Hyper-Text Transport Protocol Secure
IP	Internet Protocol
IPsec	Internet Protocol Security
IT	Information Technology
CPPND	Collaborative Protection Profile for Network Devices
MDS	Malware Defense System
MTA	Message Transfer Agent
NIT	Network iTC Interpretations Team
PP	Protection Profile
RSA	Rivest, Shamir and Adleman (algorithm for public-key cryptography)
SA	Security Association
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functions

2. TOE Description

The AhnLab MDS protection system mitigates Advanced Persistent Threat (APT) attacks by identifying known and unknown malware, detecting suspicious traffic, and removing the detected threats. The AhnLab MDS is a malware detection system that inspects network traffic and initiates intrusion mitigation. It is not a network communication filtering device (i.e., not a firewall).

The Target of Evaluation (TOE) is the AhnLab MDS, MDS with MTA, and MDS Manager V2.1.

AhnLab Malware Defense System (MDS) and MDS with MTA are network devices that provide malware detection for enterprise network security, protecting networks from Advanced Persistent Threat attacks, and provide the capability to immediately respond to remediate infected end-point systems. MDS with MTA also provides MTA functionality via purchase of separate license. MDS Manager is a network device without malware detection capabilities. In a distributed architecture, the MDS Manager Series Appliances also provide the capability to remotely manage multiple MDS and MDS/MTA appliances. However since distributed architectures are not included in the TOE, this capability is not evaluated. Each network device in the TOE: MDS, MDS with MTA, and MDS Manager provides all of the security requirements defined by the NDcPP including Security Audit, Cryptographic Support, Identification and Authentication, Security Management, Protection of the TSF, TOE Access, and Trusted path/channels.

The TOE consists of the following appliances: MDS, or MDS with MTA, or MDS Manager identified above; and the software installed on the appliance. Each TOE instance consists of a single appliance and is capable of providing all security functions specified in [NDcPP].

Internally, the devices differ in the amount of internal memory, the operating system, the type of CPU and the number of network ports. The device specifications are as follows.

Device	Main Processor	Storage	Network Ports	Operating System
MDS 1000	Intel Core i3 Dual Core	HDD: 1 TB	1G Ports(Copper) 6ea 1G Ports (Fiber) 2ea	Linux 4.18.16
MDS 2000	Intel Xeon Quad Core	HDD: 1 TB SSD: 256 GB	1G Ports(Copper) 5ea 1G Ports (Copper/Fiber combo) 4ea	Linux 4.18.16
MDS 4000	Intel Xeon 10 Core	HDD: 2TB SSD: 480GB	1GbE (Copper) 4ea 1/10G SFP+ (Fiber) 4ea	Linux 4.18.16
MDS 4000A	Intel Xeon 10 Core	HDD: 2TB	1GbE 4 Ports (Copper) 10G SFP+ 4 Ports(Optical)	Linux 4.18.16
MDS 6000	Intel Xeon Quad Core 2 each	HDD: 1TB SSD: 512GB	1G Ports(Copper) 2ea 1G Ports (Cooper) or (Fiber) 8ea 10G Ports (Fiber) 2ea	Linux 4.18.16
MDS 8000	Intel Xeon 12 Core 2 each	HDD: 4 TB SSD: 960 GB	1GbE (Copper) 4ea 1/10G SFP+ (Fiber) 4ea	Linux 4.18.16
MDS 8000A	Intel Xeon 12cores 2 each	HDD: 4 TB	1GbE 4 Ports (Copper) 10G SFP+ 4 Ports (Optical)	Linux 4.18.16

MDS10000	Intel Xeon 22 Core 2 each	HDD: 8 TB SSD: 960 GB * 2	1GbE Ports (Copper) 2ea 1/10G Base-T Ports (Copper) 4ea 1/10G SFP+ Ports (Optical) 6ea		Linux 4.18.16
MDS 10000A	Intel Xeon 20 Core 2 each	HDD: 8 TB	**Default**	**Option**	Linux 4.18.16
			1GbE 2 Ports (Copper), 10GBase-T 2Ports(Copper), 10G SFP+ 4 Ports(Optical)	1GbE 2 Ports (Copper), 10GBase-T 4 Ports (Copper), 10G SFP+ 6 Ports (Optical)	
MDS Manager 2000	Intel Xeon Dual Core 3.3 Ghz	HDD:500 GB Main: 4 GB	1G Ports(Copper) 2ea		3.14.48-gentoo (Kernel 4.18.16)
MDS Manager 5000R	Intel Xeon Quad Core 3.2 Ghz	HDD: 2 TB Main: 8 GB	1G Ports(Copper) 2ea		3.14.48-gentoo (Kernel 4.18.16)
MDS Manager 5000AR	Intel Xeon Quad Core 3.5 Ghz	HDD: 1 TB*2ea 2 TB*2ea Main: 32 GB	1G Ports(Copper) 2ea		3.14.48-gentoo (Kernel 4.18.16)
MDS Manager 10000R	Intel Xeon Quad Core 3.5 Ghz	HDD: 4 TB Main 16 GB	1G Ports(Copper) 2ea		3.14.48-gentoo (Kernel 4.18.16)
MDS Manager 10000AR	Intel Xeon 6 Core 3.6 Ghz	HDD: 2 TB*2ea 4 TB*2ea Main: 64 GB	1G Ports(Copper) 2ea		3.14.48-gentoo (Kernel 4.18.16)

2.1 TOE Overview

The MDS product is a network device providing APT analysis, detection and remediation of monitored network communications. MDS with MTA provides additional message transfer agent (MTA) functionality that is responsible for transferring and routing electronic mail messages. The MDS Manager device provides a subset of the MDS device functionality (it does not contain the analyzer component) but provides all of the security functionality specified in the [CPPND].

2.2 TOE Architecture

The TOE is comprised of one instance of the following product series appliances with software:

- MDS Series Appliance
- MDS with MTA Series Appliance
- MDS Manager Series Appliance

The specific device models and software for each series are identified in Section 1.1.

Each appliance includes a MDS Manager software component that monitors and responds to malware and abnormal traffic detected by the MDS Analyzer component (described below). The MDS Manager software component has two parts: a Data Viewer and a Host Controller. The Data Viewer records and displays logs and warnings about detected malware files and security events. The Data Viewer observes the abnormal patterns of files and network traffic transferred through the host systems within the internal network, which are logically connected to the MDS system, and controlled directly by Host Controller. The Host Controller runs threat scans and remediation commands on host systems. The Host Controller receives the commands for responding to the malware, based on the administrator's settings, and also communicates with the external MDS agent (when configured) for remediation of detected threats. MDS Agents are not included in the evaluated configuration.

The MDS Manager software component included in all appliances provide all of the security functions specified in the NDcPP: identification and authentication of administrators, auditing of security-relevant events, ability to verify the source and integrity of updates to the TOE, protected communications for administration and TOE operation, and specifies CAVP-validated cryptographic mechanisms using OpenSSL-FIPS 2.0.12.

The MDS and MDS with MTA appliances also contain the MDS Analyzer component that detects and analyzes known and unknown malicious files. The MDS Analyzer is located within the network such that it can monitor all of the network traffic. All traffic monitored by the MDS Analyzer is scanned and detected malware causes a notification to be sent to the MDS Manager software component. The MDS Analyzer does not provide any security functionality defined in this ST. The MDS with MTA appliance also consists of MTA functionality available through purchase of a separate license.

In summary, the TOE appliances are network devices composed of the following software components:

- MDS Appliance: MDS Manager software component, and MDS Analyzer component
- MDS with MTA Appliance: MDS Manager software component, MDS Analyzer component, and license to activate the MTA functionality.
- MDS Manager Appliance: MDS Manager software component

The MDS Manager software component provides all of the security functions expected for a secure network device as defined in the NDcPP. These are summarized above and in Section 2.2.2 and defined in Section 5.2 of this ST.

The TOE is a hardware appliance with embedded software installed at the factory. The TOE must be initially configured (e.g., network addresses, default routes, administrator accounts) using a command line interface from a local directly connected console.

Once network interfaces have been configured on an appliance, administration of the TOE is performed either from a local, directly connected console, or from a networked administrative workstation. Administration from a network workstations is performed using either an SSH protected terminal emulator or a TLS protected browser connection.

The AhnLab MDS products come packaged with MDS Agent software that can be subsequently installed on a host/PC to support the product’s malware detection and mitigation functions. When installed, the Agent software does not execute within the MDS appliance and does not provide any of the TOEs security functions. In the evaluated configuration, the Agent software is not installed.

The following three figures show sample deployment topologies for each of the three TOEs. Note the product supports multiple deployments and as such shows some components or multiple TOEs in a deployment. In the evaluated configuration, distributed TOEs are not supported; each TOE instance is a single device. In addition, the evaluated configuration does not include agents for any of the TOE instances. Please see the AhnLab guidance documentation for more details on the other deployment options. To assist in interpreting the figures please refer to the following:

- Solid lines: physical networks
- Dotted lines: management communication: for example the Data Viewer in the MDS Manager device can manage agent status in Host Controller in MDS or MDS Manager devices

- Red dotted line: monitoring for traffic to MDS analyzer
- Green line: MDS agent deploy
- Triangle: mirrored port

Figure 1 depicts a sample topology for the MDS Appliance TOE.

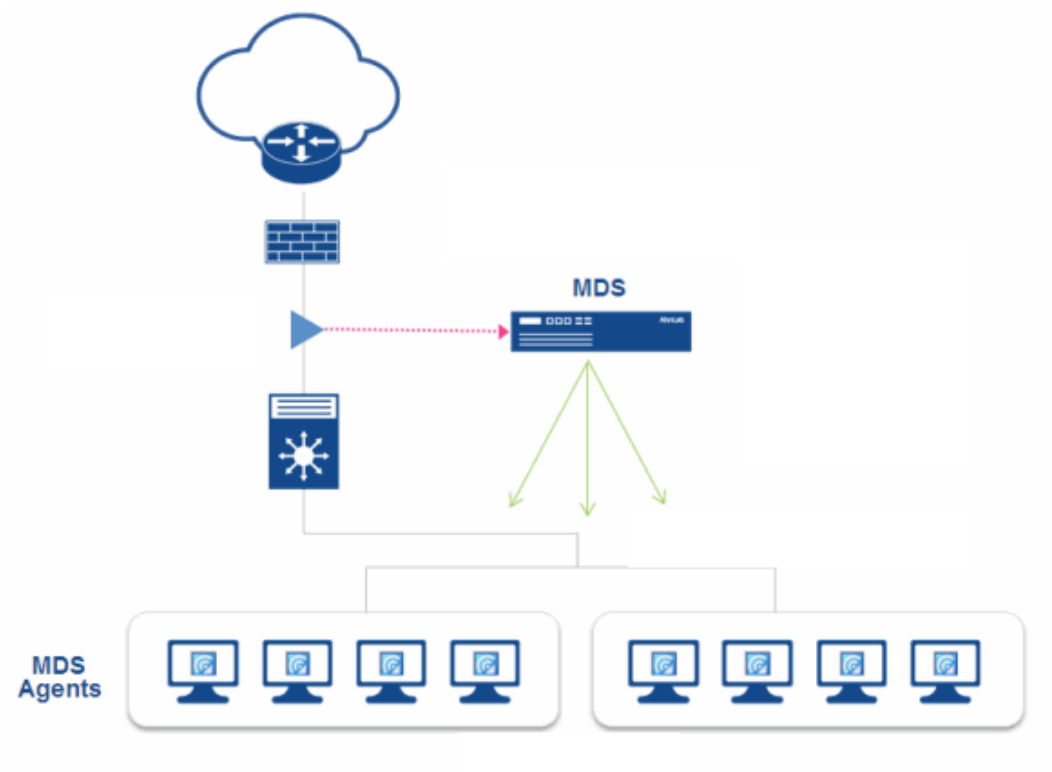

Figure 1: Sample MDS Network Topology

Figure 2 depicts a sample topology for the MDS Manager TOE. In the evaluated configuration, the MDS Manager TOE is installed as a single instance with both Data Viewer and Host Controller software and without an MDS Analyzer device. The figure depicts the MDS Managers deployed in two distinct roles – Data Viewer and Host Controller. The capabilities associated with each role are not covered by the evaluation, only the functions necessary to meet the requirements of [NDcPP].

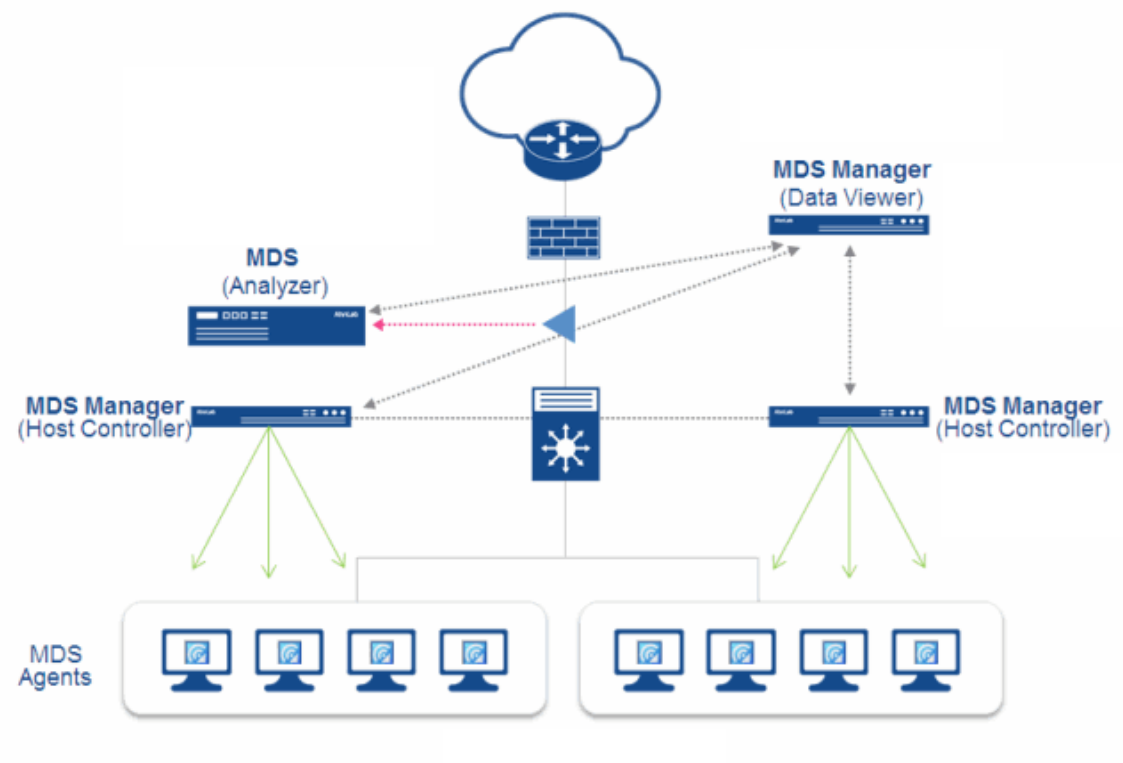

Figure 2: Sample MDS and MDS Manger Network Topology

Figure 3 depicts a sample topology for the MDS with MTA TOE.

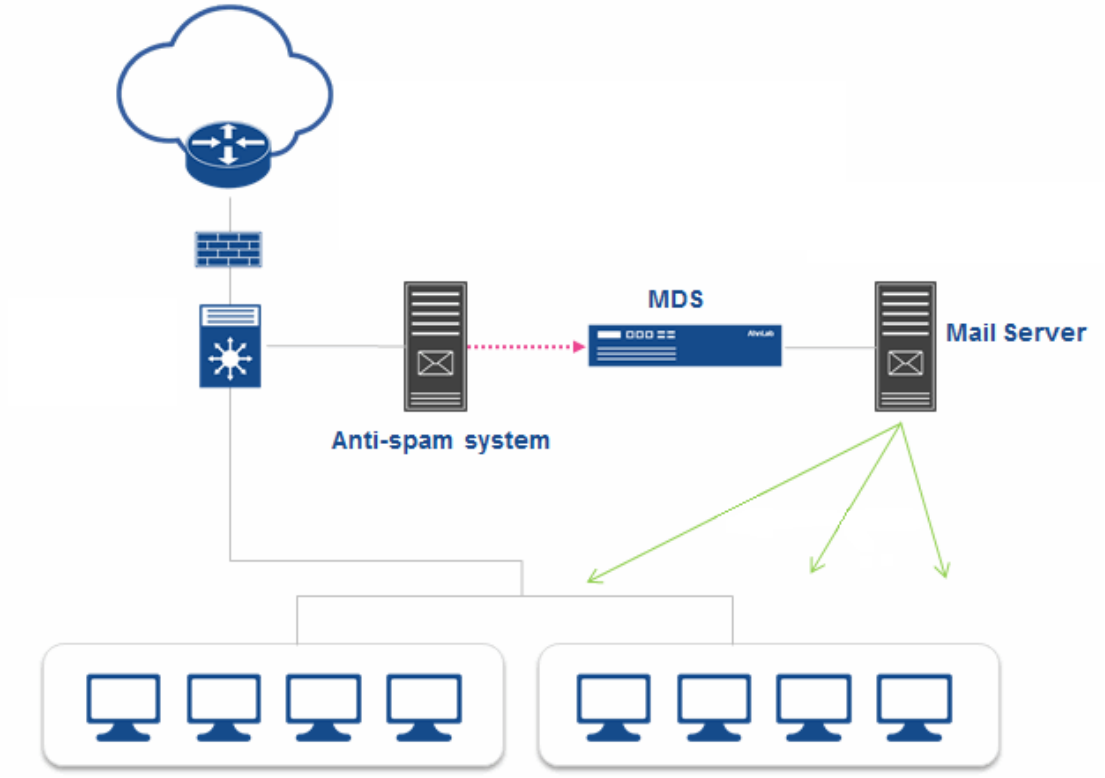

Figure 3: Sample MDS with MTA Network Topology

2.2.1 Physical Boundaries

The basic architecture of the AhnLab TOE is a single physical, hardware appliance with a local console port and network ports. The TOE is comprised of this physical appliance; and the software running within it.

Being a network device, network protocols are some of the primary interfaces to the TOE. In the evaluated configuration, the following IP ports must be configured (TOE instance dependent).

Port	Description
TCP 59005	Web Management Interface (for MDS/MDS with MTA)
TCP 58005	Web Management Interface (for MDS Manager)
TCP 22	Secure Shell connection to CLI for management
UDP 514/516	The port used to transfer the logs using the syslog port (UDP 514)

Table 1: TOE Ports

The TOE's operational environment must include:

- an external log server,
- an NTP server (optional), and
- An administrative workstation equipped with a chrome version 40 (or higher) browser and SSH client software.

SNMP is not included in the evaluated configuration and must not be configured or used. In addition, agents are not installed in the evaluated configuration.

2.2.2 Logical Boundaries

This section summarizes the security functions provided by AhnLab MDS, MDS with MTA, and Manager:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

Given that this Security Target conforms to the NDcPP, the security claims in this Security Target focus on the TOE as a secure network infrastructure device and do not focus on other key functions provided by the TOE, such as malware detection and mitigation. However, those functions can be freely used without affecting the claimed and evaluated security functions; they simply have not been evaluated to work correctly themselves.

2.2.2.1 Security Audit

The TOE generates security relevant audit records including administrative activity. The audit records are stored on the TOE, protected from unauthorised deletion and can be sent to a remote audit server for storage. The connection for transmission of audit records uses TLS.

2.2.2.2 Cryptographic Support

The TOE includes cryptographic functionality that provide random bit-generation, encryption/decryption, digital signature, secure hashing and key-hashing features. These features support cryptographic protocols including SSH, TLS and HTTPS.

2.2.2.3 Identification and Authentication

The TOE identifies and authenticates all users prior to granting them access to the Web Management or Command Line interfaces. The TOE provides the ability to define administrative accounts that have permission to view and/or modify TOE configuration variables. Each of these administrative accounts has its own password.

The TOE provides Password Management restrictions including support for minimum characters, and restrictions for character usage.

The TOE provides X.509 Certificate Validation, Authentication, and X.509 Certificate Requests for certificates used in trusted channel protocols.

2.2.2.4 Security Management

The TOE offers two administrative interfaces a Command Line Interface (CLI) provided at a local console as well as through SSH and a graphical user interface provided through TLS/HTTPS. Both interfaces require a username and password prior to allowing any administrative actions to define accounts and configure TOE functionality. SSH also supports authentication via public-key. The System Administrator is considered to be the authorized Security Administrator of the TOE (as defined in the [NDcPP]). The TOE provides functions to manage the TOE and TOE data.

2.2.2.5 Protection of the TSF

The TOE implements a number of features designed to protect itself to ensure the reliability and integrity of its security features.

The TOE ensures that sensitive information such as passwords and cryptographic keys are stored such that they are not accessible even to an administrator. The TOE provides its own internal clock which it uses to provide a reliable time source for audit records.

The TOE includes functions to perform self-tests and mechanisms for the update of the TOE software/firmware.

2.2.2.6 TOE Access

The TOE can be configured to display an informative banner when an administrator establishes an interactive session. The TOE can also enforce an administrator-defined inactivity timeout value after which the inactive session (local or remote) will be terminated. Finally, the TOE allows administrators to terminate their own session.

2.2.2.7 Trusted Path/Channels

The TOE protects administrator communications from network workstations using SSHv2, TLS v1.1 and TLS v1.2 depending upon the interface being accessed. The administrative Command Line Interface is access through the SSHv2 protocol, while TLS/HTTPS is used for the Web Management interface. In each case, both integrity and disclosure protection is ensured by the protocol being used. If the negotiation of an encrypted session fails or if the user does not have authorization for remote administration, an attempted connection is not established.

The TOE protects communication with an external log server to prevent unintended disclosure or modification of audit records.

2.3 TOE Documentation

There are numerous documents that provide information and guidance for the deployment of the TOE. In particular, the following Common Criteria specific guides reference the security-related guidance material for all devices in the evaluated configuration:

- AhnLab MDS V2.1, MDS Manager V2.1, MDS (MTA License Applied) V2.1 Configuring Common Criteria Compliance Guide v0.4 (2018.11.19)
- *AhnLab MDS Installation Guide, v1.1 (May 30, 2016)*
- *AhnLab_mds_cli_eng, v1.1 (May 30, 2016)*
- *AhnLab_MDS_Admin_Guide, v1.1 (May 30, 2016)*
- *AhnLab MDS (MTA) Installation Guide, v1.1 (May 30, 2016)*
- *AhnLab_MDS (MTA)_cli_eng, v1.1 (May 30, 2016)*
- *AhnLab MDS (MTA) Admin Guide, v1.1 (May 30, 2016)*
- *AhnLab MDS Manager Installation Guide, v1.2 (Dec 30, 2017)*
- *AhnLab_mds manager_cli_eng, v1.2 (Dec 30, 2017)*
- *AhnLab MDS Manager Admin Guide, v1.2 (Dec 30, 2017)*

3. Security Problem Definition

This security target includes by reference the Security Problem Definition (composed of organizational policies, threat statements, and assumptions) from the *collaborative Protection Profile for Network Devices Version 1.0, 27 February 2015* (CPPND). The [CPPND] offers additional information about the identified threats, but that has not been reproduced here and the [CPPND] should be consulted if there is interest in that material.

In general, the [CPPND] has presented a Security Problem Definition appropriate for network infrastructure devices, such as an IDS, and as such is applicable to the AhnLab MDS, MDS with MTA, and MDS Manager TOE.

4. Security Objectives

The [CPPND] security objectives for the operational environment are reproduced below, since these objectives characterize technical and procedural measures each consumer must implement in their operational environment. The [CPPND] does not specifically define any security objectives for the TOE.

In general, the [CPPND] has presented a Security Objectives statement appropriate for network infrastructure devices, such as an IDS, and as such is applicable to the AhnLab MDS, MDS with MTA, and MDS Manager TOE.

4.1 Security Objectives for the TOE

OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
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.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.UPDATES	The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The administrator’s credentials (private key) used to access the TOE must be protected on any other platform on which they reside

5. IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the Protection Profile (PP): *collaborative Protection Profile for Network Devices Version 1.0, 27 February 2015,* [CPPND]. As a result, refinements and operations already performed in that PP are not identified (e.g., highlighted) here, rather the requirements have been copied from that PP and any residual operations have been completed herein. Of particular note, the [CPPND] made a number of refinements and completed some of the SFR operations defined in the CC and that PP should be consulted to identify those changes if necessary.

The SARs are the set of SARs specified in [CPPND].

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the [CPPND]. The [CPPND] defines the following extended SFRs and since they are not redefined in this ST, the [CPPND] should be consulted for more information in regard to those CC extensions.

- FAU_STG_EXT.1: Protected Audit Event Storage
- FCS_HTTPS_EXT.1: HTTPS Protocol
- FCS_RBG_EXT.1: Random Bit Generation
- FCS_SSHS_EXT.1: SSH Server Protocol
- FCS_TLSC_EXT.2: TLS Client Protocol with Authentication
- FCS_TLSS_EXT.1: TLS Server Protocol
- FIA_PMG_EXT.1: Password Management
- FIA_UIA_EXT.1: User Identification and Authentication
- FIA_UAU_EXT.2: Password-based Authentication Mechanism
- FIA_X509_EXT.1: X.509 Certificate Validation
- FIA_X509_EXT.2: X509 Certificate Authentication
- FIA_X509_EXT.3: X.509 Certificate Requests
- FPT_APW_EXT.1: Protection of Administrator Passwords
- FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
- FPT_TST_EXT.1: TSF Testing
- FPT_TUD_EXT.1: Trusted Update
- FTA_SSL_EXT.1: TSF-initiated Session Locking

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User identity association
	FAU_STG.1: Protected audit trail storage
	FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM.2: Cryptographic Key Establishment (Refined)
	FCS_CKM.4: Cryptographic Key Destruction
	FCS_COP.1(1): Cryptographic Operation (AES Data Encryption/ Decryption)
	FCS_COP.1(2): Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1(3): Cryptographic Operation (Hash Algorithm)
	FCS_COP.1(4): Cryptographic Operation (Keyed Hash Algorithm)
	FCS_HTTPS_EXT.1: HTTPS Protocol
	FCS_RBG_EXT.1: Random Bit Generation
	FCS_SSHS_EXT.1: SSH Server Protocol
	FCS_TLSC_EXT.2: TLS Client Protocol with Authentication
	FCS_TLSS_EXT.1: TLS Server Protocol
FIA: Identification and authentication	FIA_PMG_EXT.1: Password Management
	FIA_UIA_EXT.1: User Identification and Authentication
	FIA_UAU_EXT.2: Password-based Authentication Mechanism
	FIA_UAU.7: Protected Authentication Feedback
	FIA_X509_EXT.1: X.509 Certificate Validation
	FIA_X509_EXT.2: X.509 Certificate Authentication
	FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	FMT_MOF.1(1)/TrustedUpdate: Management of security functions behaviour
	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_APW_EXT.1: Protection of Administrator Passwords
	FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
	FPT_TST_EXT.1: TSF Testing (Extended)
	FPT_TUD_EXT.1: Trusted Update
	FPT_STM.1: Reliable Time Stamps
FTA: TOE access	FTA_SSL_EXT.1: TSF-initiated Session Locking
	FTA_SSL.3: TSF-initiated Termination
	FTA_SSL.4: User-initiated Termination
	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF Trusted Channel (Refined)
	FTP_TRP.1: Trusted Path (Refinement)

Table 2 TOE Security Functional Components

5.2.1 Security Audit (FAU)

5.2.1.1 Audit Data Generation (FAU_GEN.1)

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 comprising:

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).*
- *Security related configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*[2]
- *Resetting passwords (name of related user account shall be logged).*
- *Starting and stopping services (if applicable)*
- ***[no other actions]**;*

d) Specifically defined auditable events listed in **Table 3**.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_SSHS_EXT.1[3]	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.2	Failure to establish a TLS session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism	Provided user identity, origin of the attempt (e.g., IP address).
FIA_X509_EXT.1	Unsuccessful attempt to validate a certificate	Reason for failure
FMT_MOF.1(1)/TrustedUpdate	Any attempt to initiate a manual update	None
FMT_MTD.1	All management activities of TSF data	None
FPT_STM.1	Changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	No additional information
FTA_SSL_EXT.1	~~Any attempts at unlocking of an interactive session.~~[4] The termination of a remote session by the session locking mechanism.	None
FTA_SSL.3	The termination of a remote session by	None

[2] There are no administrative tasks requiring generating/import of, changing, or deleting of cryptographic keys. Therefore there no audit records are generated for management of cryptographic keys.
[3] TD0150 has removed Successful SSH rekey events from the audit table (for FCS_SSHS_EXT.1).
[4] Lock the session is an optional selection in FTA_SSL_EXT.1. The SFR also allows terminate the session which is the selection for this TOE. Therefore the audit event was corrected to reflect this selection.

Requirement	Auditable Events	Additional Audit Record Contents
	the session locking mechanism.	
FTA_SSL.4	The termination of an interactive session.	None
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions	Identification of the initiator and target of failed trusted channels establishment attempt
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions	Identification of the claimed user identity

Table 3 Auditable Events

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

1) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
2) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, information specified in column three of **Table 3**.

5.2.1.2 User identity association (FAU_GEN.2)

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

5.2.1.3 Protected audit trail storage (FAU_STG.1)

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

FAU_STG.1.1 The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

5.2.1.4 Protected Audit Event Storage (FAU_STG_EXT.1)

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself.

FAU_STG_EXT.1.3 The TSF shall [***overwrite previous audit records according to the following rule***: ***[the TOE begins overwriting the oldest events when the device disk size reaches a defined percentage of its capacity]***] when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 Cryptographic Key Generation (Refined) (for asymmetric keys) (FCS_CKM.1)

FCS_CKM.1.1 The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;***

].

5.2.2.2 Cryptographic Key Establishment (Refined) (FCS_CKM.2)

FCS_CKM.2.1 The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- ***RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography";***

].

5.2.2.3 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1[5] The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [

- For plaintext keys in volatile storage, the destruction shall be executed by a ***[single overwrite consisting of [zeroes]];***
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that ***[***
  o logically addresses the storage location of the key and performs a [single] overwrite consisting of [zeroes]]

]

that meets the following: No Standard.

5.2.2.4 Cryptographic Operation (AES Data Encryption/Decryption)) (FCS_COP.1(1))

FCS_COP.1.1(1) The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC***] mode and cryptographic key sizes [***128 bits, 256 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116***].

5.2.2.5 Cryptographic Operation (Signature Generation and Verification)) (FCS_COP.1(2))

FCS_COP.1.1(2) The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm **[**

- ***RSA Digital Signature Algorithm and cryptographic key sizes (modulus)*** **[2048 bits],**

]

that meets the following: **[**

- ***For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS1v1_5[6]; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,***

].

[5] This SFR has been changed by TD0130.
[6] TD0116 has been applied to correct the type in the PP.

5.2.2.6 Cryptographic Operation (Hash Algorithm) (FCS_COP.1(3))

FCS_COP.1.1(3) The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256***] that meet the following: ISO/IEC 10118-3:2004.
Application Note: *The TOE platform uses SHA-256 for trusted update verification and for protection of Administrator passwords.*

5.2.2.7 Cryptographic Operation (Keyed Hash Algorithm) (FCS_COP.1(4))

FCS_COP.1.1(4) The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-256***] and cryptographic key sizes [**256 bits**] and message digest sizes [***256***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

5.2.2.8 HTTPS Protocol (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.
FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.
FCS_HTTPS_EXT.1.3 The TSF shall establish the connection only if [***the peer initiates handshake***].[7]

5.2.2.9 Random Bit Generation (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [***CTR_DRBG (AES)***].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[one] software-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

5.2.2.10 SSH Server Protocol (FCS_SSHS_EXT.1)

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254 and [***6668***].

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [**3500 bytes]** bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [***aes128-cbc, aes256-cbc***].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses [***ssh-rsa***] and [***no other public key algorithms***] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [***hmac-sha2-256***] and [***no other MAC algorithms***] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [***diffie-hellman-group14-sha1***] and ***[no other methods]*** are the only allowed key exchange methods used for the SSH protocol

FCS_SSHS_EXT.1.8[8] The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

[7] This requirement has been changed by TD0125.

5.2.2.11 TLS Client Protocol with Authentication (FCS_TLSC_EXT.2)

FCS_TLSC_EXT.2.1 The TSF shall implement [***TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

[***Optional Ciphersuites:***

- ***TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***].

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.2.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [***none***] and no other curves.

FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

5.2.2.12 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

[***Optional Ciphersuites:***

- ***no other ciphersuite***].

FCS_TLSS_EXT.1.2[9] The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [***none***].

FCS_TLSS_EXT.1.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [***no other size***] and [***no other***].

5.2.3 Identification and Authentication (FIA)

5.2.3.1 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***"!", "@", "#", "%", "^", "*", "(", ")", [***"_", "+", "-", "=", "[", "]", "\", "{", "}", "|", ":", ";", ",", ".", "/", "?"***]***];

b) Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater.

---

[8] This requirement has been changed by TD0167

[9] This SFR was modified by TD0156

5.2.3.2 User Identification and Authentication (FIA_UIA_EXT.1)

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [***no other actions***]

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

5.2.3.3 Password-based Authentication Mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [***none***] to perform administrative user authentication.

5.2.3.4 Protected Authentication Feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the administrative user while the authentication is in progress at the local console.

5.2.3.5 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following

rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [***Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

FIA_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

5.2.3.6 X.509 Certificate Authentication (FIA_X509_EXT.2)

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***TLS*],** and [***no additional uses***].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [***not accept the certificate***].

5.2.3.7 X.509 Certificate Requests (FIA_X509_EXT.3)

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [***Common Name, Organization, Organizational Unit, Country***].

FIA_X509_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.2.4 Security Management (FMT)

5.2.4.1 Management of Security Functions Behavior (FMT_MOF.1(1)/ TrustedUpdate)

FMT_MOF.1.1(1)/TrustedUpdate The TSF shall restrict the ability to enable the functions to perform manual update to Security Administrators.

5.2.4.2 Management of TSF Data (FMT_MTD.1)

FMT_MTD.1.1 The TSF shall restrict the ability to manage the TSF data to Security Administrators

5.2.4.3 Specification of Management Functions (FMT_SMF.1)

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

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination ~~or locking~~;
- Ability to update the TOE, and to verify the updates using [***hash comparison***[10]] capability prior to installing those updates;
- [***No other capabilities.***]

5.2.4.4 Restrictions on Security (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

- Security Administrator.

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

- The Security Administrator role shall be able to administer the TOE locally;
- The Security Administrator role shall be able to administer the TOE remotely

are satisfied.

[10] FMT_SMF.1 was modified to comply with TD0090.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 Protection of Administrator Passwords (FPT_APW_EXT.1)

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.
FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

5.2.5.2 Protection of TSF Data (for reading of all symmetric keys) (FPT_SKP_EXT.1)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.2.5.3 TSF Testing (Extended) (FPT_TST_EXT.1)

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [***during initial start-up (on power on), periodically during normal operation, at the request of the authorized user***] to demonstrate the correct operation of the TSF: [**process monitoring, software checksum tests, and cryptographic module self-tests**].

5.2.5.4 Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1 The TSF shall provide security administrators the ability to query the currently executing version of the TOE firmware/software and [***the most recently installed version of the TOE firmware/software***][11].

FPT_TUD_EXT.1.2 The TSF shall provide security administrators the ability to manually initiate updates to TOE firmware/software and [***no other update mechanism***].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [***published hash***] prior to installing those updates.

5.2.5.5 Reliable Time Stamps (FPT_STM.1)

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

5.2.6 TOE Access (FTA)

5.2.6.1 TSF-initiated Session Locking (FTA_SSL_EXT.1)

FTA_SSL_EXT.1 The TSF shall, for local interactive sessions, [
- ***terminate the session***

]
after a Security Administrator-specified time period of inactivity.

5.2.6.2 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1 Refinement: The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

5.2.6.3 User-initiated Termination (FTA_SSL.4)

FTA_SSL.4.1 Refinement: The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

[11] TD0154 has modified this SFR.

5.2.6.4 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1 Refinement: Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

5.2.7 Trusted Path/Channels (FTP)

5.2.7.1 Inter-TSF trusted channel (Refined) (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall be capable of using [***TLS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[no other capabilities]***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2 The TSF shall permit the TSF, or the authorized IT entities to initiate communication via the trusted channel.

FTP_ITC.1.3 Refinement: The TSF shall initiate communication via the trusted channel for [**transmission of audit records to the audit server**].

5.2.7.2 Trusted Path (Refinement) (FTP_TRP.1)

FTP_TRP.1.1 The TSF shall be capable of using [***SSH, HTTPS***] to provide a communication path between itself and authorized remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure and provides detection of modification of the channel data.

FTP_TRP.1.2 The TSF shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for initial administrator authentication and all remote administration actions.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference from the [CPPND].

Requirement Class	Requirement Component
ASE: Security Target	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
ADV: Development	ADV_FSP.1 Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1 Labelling of the TOE
	ALC_CMS.1 TOE CM coverage
ATE: Tests	ATE_IND.1 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.1 Vulnerability survey

Table 4 Assurance Components

Consequently, the assurance activities specified in the [CPPND] apply to the TOE evaluation.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security Audit

The TOE generates security relevant audit records including administrative activity. The audit records are stored on the TOE, protected from unauthorised deletion and can be sent to a remote audit server for storage. The connection for transmission of audit records uses TLS.

6.1.1 FAU_GEN.1: Audit Data Generation

The TOE is designed to be able to generate log records for security relevant events as they occur. The events that can cause an audit record to be logged include starting and stopping the audit function, the events identified in **Table 3**; and administrator actions comprising:

- Administrative login and logout (the log record includes the name of the user account),

- Security related configuration changes (in addition to the information that a change occurred, the TOE logs what has been changed),
- Resetting passwords (name of related user account is logged), and
- Starting and stopping services (the term 'services' refers to trusted path and trusted channel communications, on demand self-tests, trusted update and administrator sessions (that exist under the trusted path)).

There are no administrative tasks requiring generating/import of, changing, or deleting of cryptographic keys. Therefore, no audit records are generated for management for cryptographic keys.

Table 3 corresponds to the audit events specified in Table 1 of the NDcPP and includes the audit events specified in the NDcPP for optional and selected SFRs as selected in this ST. Note that the only protocol (i.e., HTTPS, TLS) failures auditable by the TOE are authentication failures for user-level connections.

6.1.2 FAU_GEN.2: User Identity Association

The logged audit records identify the date and time, the nature or type of the triggering event, an indication of whether the event succeeded or failed, and the identity of the user responsible for the event. The logged audit records also include event-specific content that includes at least all of the content required in **Table 3**.

6.1.3 FAU_STG.1: Protected audit trail storage / FAU_STG_EXT.1: Protected Audit Event Storage

The TOE includes an internal log implementation that can be used to store and review audit records locally. The local audit logs are stored on the appliance hard drive. Only authorized administrators can read the local audit trail; or delete logs from the audit trail. There are no interfaces to modify or delete individual audit records.

The TOE can be configured to send generated audit records to an external audit server using TLS v1.1 or TLS v1.2. When configured to send audit records to an audit server, audit records are also written to the external server as they are written to the local audit log.

When the local storage space for audit data is full; the TOE overwrites the oldest events. For MDS and MDS with MTA devices the TOE begins to overwrite the oldest events when the log file reaches 70% of its capacity based on disk space of the device. Each time logging occurs, the disk space is checked and if disk space is 70% full, then the TOE overwrites the oldest events with the newly generated events. For the MDS Manager device the TOE begins to overwrite the oldest events when the log file reaches an Administrator configurable percentage. The administrator can choose any percentage value from 10% to 90%. The default percentage is 80%.

6.2 Cryptographic Support

The TOE includes the AhnLab MDS OpenSSL cryptographic module: AhnLab MDS ACM, Version 1.0. The module comprises a vendor-modified version of the OpenSSL FIPS Object Module 2.0.12 linked with the FIPS-capable OpenSSL 1.0.2p and OpenSSH 7.8p1 libraries. The module provides implementations of all required cryptographic algorithms and mechanisms. The modifications were made by AhnLab to ensure the AhnLab MDS OpenSSL cryptographic module satisfies all cryptographic requirements including implementation of the FIPS 186-4 RSA Key Generation. The following functions implemented in AhnLab MDS ACM have been NIST-validated in accordance with the identified standards.

Functions	**Standards**	**CAVP Certificates**
Asymmetric key generation		
• RSA (2048-bit)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	2196

Functions	Standards	CAVP Certificates
Encryption/Decryption		
• AES CBC mode; 128, 256 bit key sizes	ISO 18033-3, CBC as specified in ISO 10116	4065
Cryptographic (Signature Generation and Verification)		
• RSA (2048 bits)	FIPS PUB 186-4 "Digital Signature Standard (DSS)"	2196
Cryptographic hashing		
• SHA-1, SHA-256	ISO/IEC 10118-3:2004	3348
Keyed-hash message authentication		
• cryptographic key sizes 256 bits and message digest sizes 256 bits	ISO/IEC 9797-2:2011	2652
Random bit generation		
• Generation: CTR_DRBG (AES) • Seed: one software-based noise source with a minimum of 256 bits of entropy	ISO/IEC 18031:2011	1218

Table 5 Cryptographic Functions

6.2.1 FCS_CKM.1: Cryptographic Key Generation (for Asymmetric Keys)

The TOE generates asymmetric cryptographic keys used for key establishment in accordance with FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3. See table above for Asymmetric key generation: RSA (2048-bit).

6.2.2 FCS_CKM.2: Cryptographic Key Establishment (Refined)

The TOE performs cryptographic key establishment in accordance with NIST Special Publication 800-56B. The TOE acts as both a sender and as a recipient for RSA-based key establishment schemes. See Table 5 for detail. above.

6.2.3 FCS_CKM.4: Cryptographic Key Destruction

The TOE uses the following secret keys, private keys and CSPs.

Key/CSP Name	Algorithm/Key Size	Description
RSA SGK	RSA 2048 bits	RSA signature generation key
RSA KDK	RSA 2048 bits	RSA key decryption (private key transport) key
AES EDK	AES 128, 256 bits	AES encrypt/decrypt key
HMAC Key	HMAC 256 bits	HMAC keyed hash key
CTR_DRBG Key	AES 256 bits	Internal CTR_DRBG key variable

Table 6 SFR Protection Profile Sources

The TOE incorporates OpenSSL, which provides implementation of the cryptographic algorithms specified in Table 5. The TOE invokes the OpenSSL cryptomodule APIs to set up and maintain the full TLS/SSH sessions, using the underlying cryptographic algorithms as identified in Table 5. Therefore, all key generation, negotiation of session keys, and packet authentication is performed by the cryptomodule. All secret keys, plaintext private keys, and CSPs (see Table 6 above) are managed by the cryptomodule and stored in plaintext in RAM. The cryptomodule stores

CTR_DRBG state values for the lifetime of the DRBG instance and destroys them when the DBRG is uninstantiated. The cryptomodule does not store any other secret or private keys or CSPs persistently (beyond the lifetime of an API call). They are destroyed automatically by the API when no longer required by overwriting once with 0s.

The cryptographic keys in volatile memory are destroyed as follows. The destruction is executed by a single direct overwrite consisting of zeroes: followed by a read-verify. If the read-verification of the overwritten data fails, the process is repeated again.

The TOE stores the Certificate files, CA-Certificate files, Private-Key files, and CRL files used in communication between TOE components in encrypted PEM format. The Certificate, CA Certificate, Private-Key and CRL files used for user communication with the TOE and for TOE to trusted external IT entities (Syslog Server) are never loaded or stored in memory by the AhnLab code. The files are stored on the file system and in all cases the files are passed to OpenSSL via API calls that pass in the complete filename including full path. Each API call return is checked to make sure there were no errors. The cryptomodule itself does not return sensitive data values and is responsible for ensuring the memory that held those file contents gets zeroized.

6.2.4 FCS_COP.1(1): Cryptographic Operation (Data Encryption/Decryption)

The TOE performs 128/256-bit AES encryption/decryption in CBC accordance with ISO 18033-3 and ISO 10116]. See also table above.

6.2.5 FCS_COP.1(2): Cryptographic Operation (Signature Generation and Key Verification)

The TOE performs RSA (2048-bits) Digital Signature Generation and Key Verification in accordance with FIPS PUB 186-4. See also table above.

6.2.6 FCS_COP.1(3): Cryptographic Operation (Hash Algorithm)

The TOE performs SHA-1 cryptographic hashing services in accordance with ISO/IEC 10118-3:2004. The SHA hash algorithm is used as part of HMAC, but is also used as part of RSA digital signature creation and verification. SHA-256 is used to verify the trusted update images and for protection of Administrator passwords.

6.2.7 FCS_COP.1(4): Cryptographic Operation (Keyed Hash Algorithm)

The TOE performs HMAC-SHA-256 keyed-hash message authentication with 256-bit cryptographic key, block size 512 bits and message digest sizes of 256-bits that meet the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.

6.2.8 FCS_RBG_EXT.1 Random Bit Generation

The TOE implements a CTR_DRBG(AES) random bit generator to support generation of symmetric keys and asymmetric key pairs; and in accordance with ISO/IEC 18031:2011. The deterministic RBG is seeded by one entropy source that accumulates entropy from one software-based noise source with a minimum of 256 bits of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it generates. See also table above.

6.2.9 FCS_SSHS_EXT.1 SSH Protocol

The TOE acts as an SSH Server for accepting SSH connections for accessing the CLI from authorized administrators. The TOE implements the SSHv2 protocol and complies with RFCs 4251, 4252, 4253, 4254, and 6668.

The TOE’s SSH protocol implementation supports public-key-based, and password-based authentication methods as described in RFC 4252. The TOE drops packets greater than 3500 bytes in an SSH transport connection as described in RFC 4253. As SSH packets are being received, the TOE uses a buffer to build all packet information. Once

complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full (3500 bytes) the packet will be dropped.

The TOE's SSH transport implementation uses:

- aes128-cbc, and aes256-cbc encryption algorithms
- ssh-rsa as its public key algorithm; and
- hmac-sha2-256 as its MAC algorithm.

The TOE rejects all other encryption, public key and MAC algorithms not listed above.

The TSF uses diffie-hellman-group14-sha1 key exchange method for the SSH protocol; and ensures that the SSH connection is rekeyed when a threshold of either one hour has been reached, or when one gigabyte of data has been transmitted. Both thresholds are checked by the TOE and rekeying is performed upon reaching the threshold whichever is hit first.

6.2.10 FCS_TLSC_EXT.2: TLS Client Protocol with Authentication

The TOE acts as a TLS client for secure connections with an external audit server. The TOE verifies that the presented identifier matches the reference identifier according to RFC 6125; and only establishes a trusted channel if the peer certificate is valid. The TOE determines certificate validity by verifying the identifier, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

The TOE includes support for client-side certificates for TLS mutual authentication using X509v3 certificates. The X509v3 certificates used are discussed in FIA_X509_EXT.2.1.

The TOE uses TLS 1.2 in accordance with RFC 5246 and supports the ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268

During initial configuration, the administrator configures the TOE to communicate with a designated external audit server using ip addresses; and generates a new X509 certificate that is subsequently installed on the audit server along with the CA certificate and CRL.

The TOE compares the audit server's presented identifier to the reference identifier by matching the certificate Common Name, Organization, Organizational Unit and Country against the configured server. The TOE supports the following distinguished name attributes: Common Name; Organization; Organizational Unit; and Country.

attributes	values
Common Name	AhnLab
Organization	AhnLab Inc.
Organizational Unit	Department of R&D
Country	KR

IP address reference identifiers, wildcards, and certificate pinning are not supported.

6.2.11 FCS_HTTPS_EXT.1

The TOE uses HTTPS when remote administrators connect to the TOE's GUI. The TOE's HTTPS protocol complies with RFC 2818 and is implemented using TLS 1.2 (RFC 5246). Note that the TOE does not support mutual authentication for this communication channel and therefore no peer certificate is presented. Peer authentication is performed via username and password credentials.

6.2.12 FCS_TLSS_EXT.1: TLS Server Protocol

The TOE acts as a TLS Server when remote administrators connect to the TOE's GUI using HTTPS. The TOE's HTTPS protocol complies with RFC 2818 and is implemented using TLS 1.2 (RFC 5246) and TLS 1.1 (RFC 4346) supporting the following the ciphersuite:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

The key agreement parameters of the server key exchange message consist of the following key establishment parameters generated by the TOE: RSA with key size 2048 bits.

The TOE denies connections from clients (admins) requesting connections using SSL 2.0, SSL 3.0, or TLS 1.0.

6.3 Identification and Authentication

The TOE identifies and authenticates all users prior to granting them access to the Web Management or Command Line interfaces. The TOE provides the ability to define administrative accounts that have permission to view and/or modify TOE configuration variables. Each of these administrative accounts has its own password.

The TOE provides Password Management restrictions including support for minimum characters, and restrictions for character usage.

The TOE provides X.509 Certificate Validation, Authentication, and X.509 Certificate Requests for certificates used in trusted channel protocols.

6.3.1 FIA_PMG_EXT.1 Password Management

The TOE can be composed of passwords from any combination of upper and lower case letters, numbers, and the following special characters: "!", "@", "#", "%", "^", "*", "(", ")", "_", "+", "-", "=", "[", "]", "\", "{", "}", "|", ":", ";", ",", ".", "/",and"?". A minimum password length of 15 characters can be set by the Security Administrator. Password composition rules include: the password must not contain adjacent numbers or alphabets such abcde, edcba, 12345 or 54321, and must not consist of 3 or more repeated characters such as aaa or 111.

6.3.2 FIA_UIA_EXT.1: User Identification and Authentication, FIA_UAU_EXT.2: Password-based Authentication Mechanism. FIA_UAU.7: Protected Authentication Feedback

The TOE provides a local password-based authentication mechanism and requires users to be identified and authenticated before they can access any of the TOE functions. The only capabilities allowed prior to users authenticating are the display of the warning banner before authentication.

Administration of the TOE is performed either from a local, directly connected console, or from a networked administrative workstation. Administration from a network workstation is performed using either an SSH protected terminal emulator or a TLS protected browser connection. From a desktop PC or laptop with a RS232 (DE-9) serial port, the administrator runs a terminal emulator program, such as Putty, SecureCRT or HyperTerminal. Administrators can also log in to the CLI using a remote SSH connection using a terminal emulation program and port TCP 22 to make the connection. The administrator can logon to the GUI by using a secure connection (https) from a web browser and entering the IP address of the TOE. The administrator must enter the administrator's ID and password to connect to the CLI or GUI. Remote connections using SSH can alternatively use public key-based authentication for log-in to the CLI. A successful log-in will display the command-line prompt: "MDS#" or the GUI. The TOE provides only obscured feedback to the administrative user while the authentication is in progress at the

local console. Note: for improved security, security administrators must change the default password after initial log in.

6.3.3 FIA_X509_EXT.1: X509 Certificate Validation

The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS connections with the external audit server. The TOE validates revocation status of the certificate using a Certificate Revocation List (CRL) as specified in RFC 5759.

The TOE validates the certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for the CA certificate. The certificate path must terminate with a trusted CA certificate for the connection to be established.

The TOE uses a CRL to check the validity of the certificate. The TOE checks the status of the certificate with the list. Certificate revocation status is checked after receiving the certificate from the syslog server during a TLS connection attempt. If it is revoked and listed in a CRL, the certificate is not valid. When the TOE cannot establish a connection to determine the validity of a certificate, the TOE allows the administrator to choose whether to accept the certificate.

The TOE uses the following rules for validating the extendedKeyUsage field:

- Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
- Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

The TOE supports the following distinguished name attributes: Common Name; Organization; Organizational Unit; and Country.

attributes	values
Common Name	AhnLab
Organization	AhnLab Inc.
Organizational Unit	Department of R&D
Country	KR

6.3.4 FIA_X509_EXT.2: X509 Certificate Authentication

The TOE uses X.509v3 certificates in conformance with RFC 5280 to support authentication for TLS connections with the external audit server.

During initial configuration, the administrator configures the TOE to communicate with a designated external audit server using ip address and port; and generates new X509 certificate for the TOE and audit server. The audit server certificate is subsequently installed on the audit server along with the CA certificate, and CRL. The TOEs certificate is stored in a trusted store of the device (the trusted root CA).

When a connection cannot be established during the validity check of a certificate used in establishing a trusted channel; the TOE will not accept the certificate.

6.3.5 FIA_X509_EXT.3: X509 Certificate Requests

The TOE generates a Certificate Request Message as specified by RFC 2986 and is able to provide the following information in the request: Common Name, Organization, Organizational Unit, and Country.

6.4 Security Management

The TOE provides CLI and GUI management interfaces to support security management of the TOE. The CLI is accessible via direct connection to the management port on the device, or remotely over SSH. The GUI is accessible remotely using HTTPS. The management interfaces enable the authorized administrators to configure the TOE functions and to manipulate TOE data.

The TOE supports three types of administrator accounts:

• Read-only Administrator: Administrator with read-only privileges to access the administrative interfaces.

• System Administrator: Administrator with full privileges to access and manage the entire system.

•Policy Administrator: Administrator with restricted privileges to access to some menus and run commands.

The System Administrator is considered to be the authorized Security Administrator of the TOE (as defined in the [NDcPP]. There are no non-administrative users of the TOE. Security Administrators can administer the TOE both locally and remotely.

6.4.1 FMT_MOF.1(1)/TrustedUpdate: Management of security functions behaviour

The initiation of manual TOE updates is restricted to Security Administrators.

6.4.2 FMT_MTD.1: Management of TSF Data

The Security Administrator is responsible for resetting passwords for other administrators. Administrators that are not Security Administrators cannot change their own password. No administrative functions are accessible prior to administrator log-in.

6.4.3 FMT_SMF.1: Specification of Management Functions

The TOE is capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination; and
- Ability to update the TOE, and to verify the updates using hash comparison capability prior to installing those updates;

6.4.4 FMT_SMR.2: Restrictions on Security Roles

The TOE maintains the Security Administrator role that can be assigned to users. The Security Administrator role is able to administer the TOE locally and remotely.

6.5 Protection of the TSF

The TOE ensures that sensitive information such as passwords and cryptographic keys are stored such that they are not accessible even to an administrator. The TOE provides its own internal clock which it uses to provide a reliable time source for audit records.

The TOE includes functions to perform self-tests and mechanisms for the update of the TOE software/firmware.

6.5.1 FPT_APW_EXT.1: Protection of Administrator Passwords

Administrator password authentication data are stored encrypted using SHA256 and the TOE does not provide any interfaces to read plaintext passwords.

6.5.2 FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)

Please see Section 6.2.3 above for a description of the how the preshared keys, symmetric keys, and private keys are stored. The TOE does not offer any functions that will disclose to any users a stored cryptographic key.

6.5.3 FPT_TST_EXT.1: TSF Self Testing (Extended)

The TOE includes a number of built in self-tests that are run during start-up and periodically during normal operation to determine whether the TOE is operating properly. The built-in self-tests include basic process watching (not memory), software checksum tests, and cryptographic module self-tests. The TOE reboots and logs any failures, when an error is encountered. The TOE's startup process: process_manager, is responsible for bringing up all relevant MDS processes; and provides basic process watching (that is, each process is started as expected). The TOE will automatically attempt to re-start any failed process at a one second interval. All binaries include an embedded integrity verification SHA2 checksum that is verified by the Linux daemon crontab at startup and periodically at 1 hour intervals. The administrator can manually execute the software checksum test for the Analyzer, Data Viewer, and Host Controller TOE components using the check_integrity and integrity check commands. The TOE includes CAVP certified OpenSSL binaries which are included in the self-testing to ensure the correct operation of cryptographic functions. If any of the checks fail, when the TOE reboots manually, the binaries are replaced automatically with backed-up TOE recovery configuration files. In addition, a description of the self-test failure is logged and available for Administrators to review.

6.5.4 FPT_TUD_EXT.1: Trusted Update

The TOE provides graphical user interfaces for administrators to update the TOE, and to query both the currently executing software version of the TOE as well as the most recently installed software version of the TOE.

Customers are notified by email when a firmware update is available. The TOE update file (which has been hashed using SHA-256) and the file containing the hash value are hand delivered to the end user. In Korea, both files are hand delivered to the end user by an AhnLab representative. Outside of Korea, the files are downloaded from the AhnLab File Support System (via HTTPS/SSL) by a local partner who then hand delivers them to the end user. The authorized administrator logs onto the TOE, identifies the location of the two files and then selects to update the TOE. If the authorized administrator does not elect to update the TOE at this time, the most recently installed software version will not replace the currently executing TOE version until the TOE is rebooted. Prior to the update, the TOE compares the hash of the candidate update with the hash file. If the two do not agree, the TOE refuses the update, otherwise the update is performed.

6.5.5 FPT_STM.1: Reliable Time Stamps

The TOE is a hardware appliance that includes a hardware-based real-time clock. The TOE's embedded OS manages the clock and exposes administrator clock-related functions. The Administrator can either configure an NTP Server or make updates to time using the *Direct Input* selection under the *Settings* tab. The clock is used for audit record time stamps, measuring session activity for termination, and for cryptographic operations based on time/date.

6.6 TOE Access

The TOE can be configured to display an informative banner when an administrator establishes an interactive session. The TOE can also enforce an administrator-defined inactivity timeout value after which the inactive session (local or remote) will be terminated. Finally, the TOE allows administrators to terminate their own session.

6.6.1 FTA_SSL_EXT.1: TSF-Initiated Session Locking

The TOE can be configured by an administrator to set an interactive session timeout value (any integer value greater than zero in minutes) for local user sessions. The default timeout is 10 minutes and the feature can be disabled by setting the value to 0. A local session that is inactive for the defined timeout period will be terminated. The user will

be required to re-enter their user id and their password so they can establish a new session once a session is terminated. If the user id and password match those of the user that was locked, the session is reconnected and normal input/output can again occur for that user.

6.6.2 FTA_SSL.3: TSF-initiated Termination

The TOE can be configured by an administrator to set an interactive session timeout value (any integer value greater than zero in minutes) for remote user sessions. The default timeout is 10 minutes and the feature can be disabled by setting the value to 0. A remote session that is inactive (i.e., no commands issuing from the remote client) for the defined timeout value will be terminated. The user will be required to re-enter their user id and their password so they can establish a new session once a session is terminated. If the user id and password match those of the user that was locked, the session is reconnected and normal input/output can again occur for that user.

6.6.3 FTA_SSL.4: User-initiated Termination

An administrator can terminate their own session by entering the exit command at the CLI command prompt "hostname#>". From the GUI, and administrator can terminate their own session by clicking on the "*Logout*" button at the upper right-hand corner in the common menu.

6.6.4 FTA_TAB.1: Default TOA Access Banners

Before establishing an administrative user session, the TOE displays a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE. The banner is displayed locally (serial port) and remotely (both HTTPS and SSH).

6.7 Trusted Path/Channels

The TOE protects administrator communications from network workstations using SSHv2, TLS v1.1 or TLS v1.2 depending upon the interface being accessed. The administrative Command Line Interface is access through the SSHv2 protocol, while TLS/HTTPS is used for the Web Management interface. In each case, both integrity and disclosure protection is ensured by the protocol being used. If the negotiation of an encrypted session fails or if the user does not have authorization for remote administration, an attempted connection is not established.

The TOE protects communication with an external log server to prevent unintended disclosure or modification of audit records.

6.7.1 FTP_ITC.1: Inter-TSF trusted channel

The TOE can be configured to export audit records to an external audit server. The TOE uses TLS v1.2, to protect communications between itself and the audit server. The TOE initiates communication via the trusted channel for the audit server.

The TOEs secure protocols are supported by NIST-validated cryptographic mechanisms included in the TOE implementation.

6.7.2 FTP_TRP.1: Trusted Path

To support secure remote administration, the TOE includes implementations of HTTPS and SSHv2. An authorized administrator can establish secure remote connections with the TOE using HTTP over TLS or SSHv2. When administrators attempt to connect to the TOE using TLS or SSH, the TOE negotiates cryptographic parameters for the session. If negotiation is not successful, the connection is dropped and no session is created. If the negotiation is successful, the TOE performs identification and authentication of the administrator. To successfully establish an interactive administrative session, the administrator must be able to provide acceptable user credentials (e.g., user id and password), after which they will be able to access the CLI or GUI features.

7. Protection Profile Claims

This ST is conformant to the *collaborative Protection Profile for Network Devices Version 1.0, 27 February 2015* and including the following optional SFRs: FAU_STG.1, FCS_HTTPS_EXT.1, FCS_SSHS_EXT.1, FCS_TLSC_EXT.2, and FCS_TLSS_EXT.1.

As explained in Section 3, Security Problem Definition, the Security Problem Definition of the [CPPND] has been included by reference into this ST.

As explained in Section 4, Security Objectives, the Security Objectives of the [CPPND] has been included by reference into this ST.

The SFRs in this ST are reproduced from [CPPND] and assignment and selection operations completed as appropriate, with the following variations:

- FMT_SMF.1.1 has been refined to remove “or locking” since the TOE does not lock the session but rather terminates the session as indicated by the selection in FTA_SSL_EXT.1.1.

8. Rationale

This security target includes by reference the [CPPND] Security Problem Definition, Security Objectives, and Security Assurance Requirements. The security target makes no additions to the [CPPND] assumptions. [CPPND] security functional requirements have been reproduced with the Protection Profile operations completed. Operations on the security requirements follow [CPPND] application notes and assurance activities. Consequently, [CPPND] rationale applies but is incomplete. The TOE Summary Specification rationale below serves to complete the rationale required for the security target.

8.1 TOE Summary Specification Rationale

Each subsection in Section 6, the TOE Summary Specification, describes a security function of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The set of security functions work together to satisfy all of the security functions and assurance requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification are all necessary for the required security functionality in the TSF. **Table 7 Security Functions vs. Requirements Mapping** demonstrates the relationship between security requirements and security functions.

	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FAU_GEN.1	X						
FAU_GEN.2	X						
FAU_STG.1	X						
FAU_STG_EXT.1	X						
FCS_CKM.1		X					
FCS_CKM.2		X					
FCS_CKM.4		X					
FCS_COP.1(1)		X					
FCS_COP.1(2)		X					
FCS_COP.1(3)		X					
FCS_COP.1(4)		X					
FCS_HTTPS_EXT.1		X					
FCS_RBG_EXT.1		X					
FCS_SSHS_EXT.1		X					
FCS_TLSC_EXT.2		X					
FCS_TLSS_EXT.1		X					
FIA_PMG_EXT.1			X				
FIA_UIA_EXT.1			X				
FIA_UAU_EXT.2			X				
FIA_UAU.7			X				
FIA_X509_EXT.1			X				
FIA_X509_EXT.2			X				
FIA_X509_EXT.3			X				
FMT_MOF.1.1(1)/TrustedUpdate				X			
FMT_MTD.1				X			
FMT_SMF.1				X			
FMT_SMR.2				X			
FPT_APW_EXT.1					X		
FPT_SKP_EXT.1					X		

	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FPT_TST_EXT.1					X		
FPT_TUD_EXT.1					X		
FPT_STM.1					X		
FTA_SSL_EXT.1						X	
FTA_SSL.3						X	
FTA_SSL.4						X	
FTA_TAB.1						X	
FTP_ITC.1							X
FTP_TRP.1							X

Table 7 Security Functions vs. Requirements Mapping