ZombieZERO Inspector V4.0

Security Target Lite

V1.10

April 12, 2019

* This document was evaluated in the form of Korean version, and translated into English version

Document History

Version	Revised Content	Date
v1.0	▪ First written by	April 3, 2018
V1.1	▪ Modification by OR	July 5, 2018
V1.2	▪ Modification by OR	Dec. 19, 2018
V1.3	▪ TOE Overview and description modification	Jan. 17, 2019
V1.4	▪ Modification by OR	Feb. 11, 2019
V1.5	▪ Modification by OR	Feb. 25, 2019
V1.6	▪ Modification by OR	March 8, 2019
V1.7	▪ Modification by OR	March 3, 2019
V1.8	▪ Adding Cryptographic Support SFRs	March 21, 2019
V1.9	▪ Modification of security statement for operating environment, etc.	March 27, 2019
V1.10	▪ Sanitized version of the ST V1.10	April 12, 2019

Table of Contents

List of tables

1. Introduction of Security Target

This chapter introduces the Security Target of 'ZombieZERO Inspector V4.0' of NPCore, Inc..

1.1 Security Target Reference

[Table 1] Security Target Reference

Security Target Title	ZombieZERO Inspector V4.0 Security Target Lite
Security Target Version	V1.10
Security Target Written by	NPCore, Inc.
Written Date	April 12, 2019

1.2 TOE Reference

[Table 2] TOE Reference

TOE Title	ZombieZERO Inspector V4.0
TOE Version	V4.0 Revision 12
TOE Components	ZombieZERO Detector V4.2.71 Analyzer Agent V4.0.0 ZombieZERO Agent V4.0.227
Guidance Documents	ZombieZERO Inspector V4.0 Installation Manual V1.4 ZombieZERO Inspector V4.0 Manual for Administrator V1.4
Developer / Sponsor	NPCore, Inc.

1.3 TOE Overview

ZombieZERO Inspector V4.0(hereinafter, TOE) is a software-type security product that detects malicious code coming from outside and blocks or isolates it based on pattern-based and behavioral analysis technique and it blocks internal users to access in unauthorized IP/URL (ex. malicious code distribution site).
It consists of ZombieZERO Detector V4.2.71 (hereafter referred to as ZombieZERO Detector), Analyzer Agent V4.0.0 (hereafter referred to as Analyzer Agent), and ZombieZERO Agent V4.0.227 (hereafter referred to as ZombieZERO Agent).

The major security characteristics of the TOE are as follows.

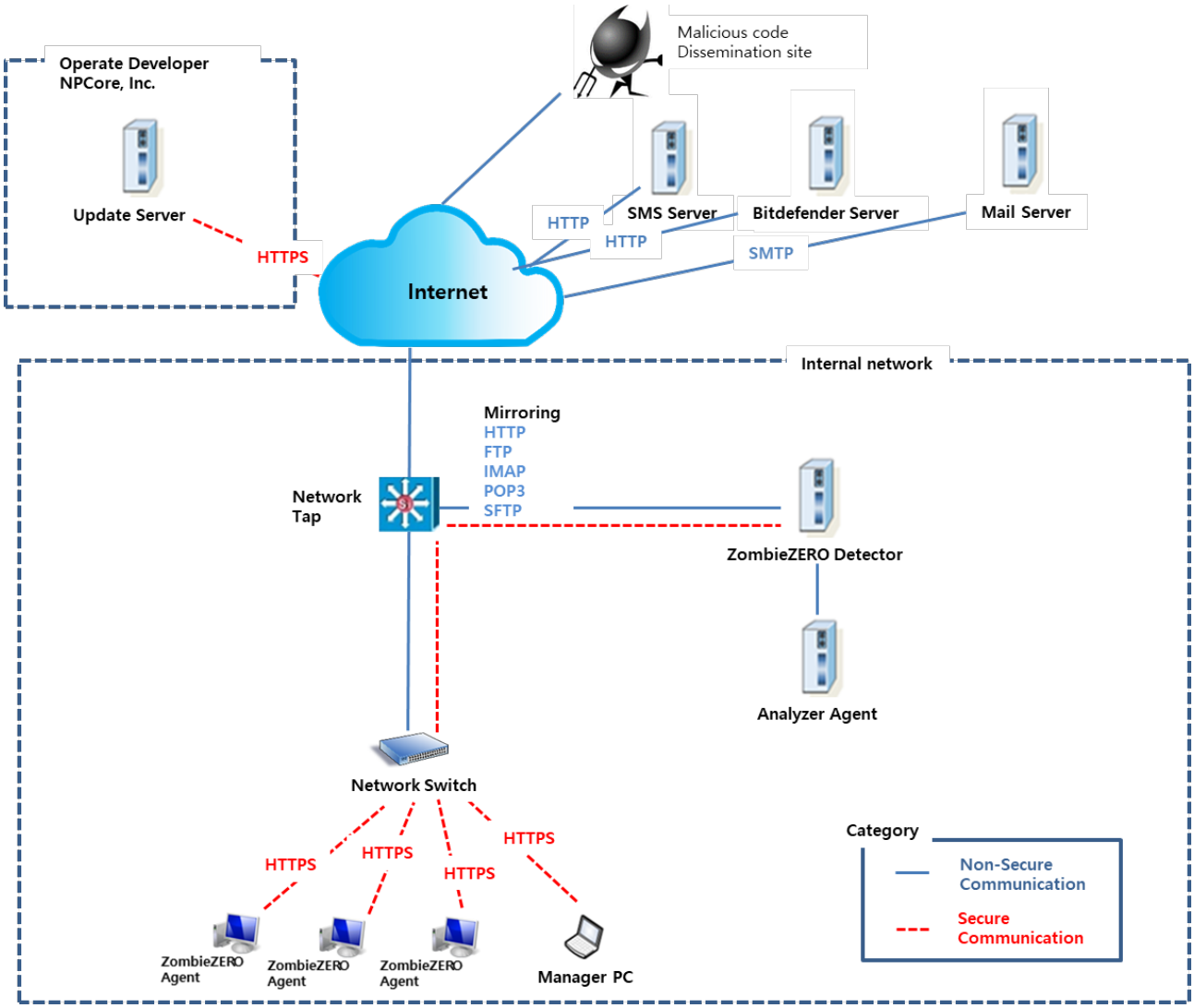
- ZombieZERO Detector
  - File collection/storage: It collects and stores files among traffic coming from the Internet through the HTTP / 1.1, FTP, POP3, IMAP, and SMTP protocol and exe files sent from the ZombieZERO Agent.
  - Information analysis: The information analysis is performed on collected files for malicious code detection as follows.
    - ✓ Pattern analysis: Malicious codes are detected based on the signature pattern and the detection rule in files collected through network protocol and files that received the analysis request from ZombieZERO Agent.
    - ✓ Behavior analysis: Malicious codes are detected by comparing the file behavior information Analyzer Agent collected with detection rules.
  - Reaction: Malicious codes detected by file information and security policy are transmitted to ZombieZERO Agent.
  - Information flow block: It blocks information flow when a packet is transmitted from an internal user PC to a known IP / URL (ex. malicious code distribution sites).
  - Security management: It provides security management functions such as security policy setting, audit record, pattern management through administrator web page (GUI).
  - in addition, security audit function, identification and authentication function, cryptographic support function, TSF protection function, TOE access control function are provided.
- Analyzer Agent
  - Behavior information collection : It executes the malicious code suspected file received from the ZombieZERO Detector to collect the file behavior, process behavior, registry behavior, network behavior and memory behavior of the file, and transmits the collected information to the ZombieZERO Detector.
- ZombieZERO Agent
  - Reaction : Files that contain detected malicious codes transferred from ZombieZERO Detector are blocked and isolated from the internal users' PCs according to the security policy as follows:
    - ✓ Blocking malicious code: Files that contain the malicious code transferred from ZombieZERO Detector are suspended the execution according to the security policy.

- ✓ Isolating malicious code: Files that contain the malicious code transferred from ZombieZERO Detector are suspended the execution and moved to an isolated space according to the security policy

- In addition, Cryptographic support function, TSF protection functions are provided.

1.3.1 Operational Environment of TOE

The operating environment of the TOE is shown in [Figure 1]



[Figure 1] Operational Environment of ZombieZERO Inspector V4.0

- ZombieZERO Detector is connected to a network terminal access point (TAP) installed at a place where external and internal networks are connected, and receives port-mirrored traffic through the Network TAP.
- Analyzer Agent installed and operated on a guest OS is not connected to other external IT entities via the network, but directly connected to ZombieZERO Detector and operated.
- ZombieZERO Agent is installed and operated in internal users' PCs.
- Encryption communication between ZombieZERO Detector and ZombieZERO Agent and between an administrator PC and ZombieZERO Detector for security management are based on the Transport Layer Security (TLS) 1.2(TLS Cipher Suites in Windows 10 v1607) protocol provided by IIS10.0 installed in ZombieZERO Detector.
- An update server run by the developer NPCore Inc. provides used up-to-date information owned by the developer, and transmits it to using TLS 1.2 protocol provided by the update server to updated files to ZombieZERO Detector after electronic signature. Commercial anti-virus server provides updates to the signature pattern using the TLS 1.2 protocol provided by the update server.

1.3.2 Non-hardware/software required by the TOE

The system requirements to run the TOE are as follows

Classification	Item	Minimum system requirement
ZombieZERO Detector	CPU	▪ Intel Xeon Quad Core 2.13 GHz or higher * 1 item
	RAM	▪ 4GB or higher
	HDD	▪ Available space of more than 1 GB required to install ZombieZERO Detector
	NIC	▪ 10/100/1000Mbps * 4 port
	OS	▪ Windows Server 2016 Standard 64bit
	SW	▪ MySQL Community Server 5.6.43 ▪ IIS 10.0 ▪ VMware VIX API 1.14.0
Analyzer Agent	CPU	▪ Intel Xeon Quad Core 2.13 GHz or above * 1 item
	RAM	▪ 4GB or higher
	SSD	▪ Available space of more than 50MB required to install Analyzer Agent
	NIC	▪ 10/100/1000Mbps * 1 port
	Hypervisor (Type1)	▪ VMware ESXi 5.5.0 U3
	Guest OS (4 items)	▪ Windows 7 Professional 32bit ▪ Windows 7 Professional 64bit ▪ Windows 10 Pro 32bit ▪ Windows 10 Pro 64bit
	SW (Installed on the Guest OS)	▪ .net framework 4.6 ▪ Microsoft Office 2013 ▪ Adobe Acrobat pro XI ▪ Adobe Flash Player 16 ▪ Hancom Office 2010
ZombieZERO Agent	CPU	▪ Intel Celeron 2.6 GHz or above * 1 item
	RAM	▪ 4GB or higher
	HDD	▪ Available space of more than 50MB required to install ZombieZERO Agent
	NIC	▪ 10/100/1000Mbps * 1 port
	OS	▪ Windows 7 Professional 32bit ▪ Windows 7 Professional 64bit ▪ Windows 10 Pro 32bit ▪ Windows 10 Pro 64bit

- An environment where Analyzer Agent is evaluated is the one where four guest OSs are concurrent run.
- A hardware where the ZombieZERO Detector is operated needs following four communication ports.
  - 2 mirroring ports (each of mirroring packets from external to internal networks and from internal to external networks)
  - 1 network communication port
  - 1 port for direct communication between ZombieZERO Detector and Analyzer Agent

The system requirements of administrator PC for security management are as follows

Administrator PC	CPU	▪ Intel Celeron 2.6 GHz or above
	RAM	▪ 4GB or higher
	HDD	▪ 500GB or higher
	NIC	▪ 10/100/1000Mbps * 1 port or more
	OS	▪ Windows 7 Professional 32bit ▪ Windows 7 Professional 64bit ▪ Windows 10 Pro 32bit ▪ Windows 10 Pro 64bit
	Web Browser	▪ Internet Explorer 10 ▪ Internet Explorer 11
	SW	▪ VMware vSphere Client 5.5 (Used when the guest OS of Analyzer Agent is installed.)

The TOE provides interworking functions with various external systems according to detailed options of the security policy (feature) defined by the authorized administrator and normal operations. The following software is the operational environment of the TOE and out of the scope of the TOE.

MySQL Community Server 5.6.43	ZombieZERO Detector uses MySQL Community Server 5.6.40, which is a database management system (DBMS), to store audit data generated during the TOE operation.
IIS 10.0	ZombieZERO Detector uses IIS 10.0 to provide web-based security management, and also employs the TLS Cipher Suites in Windows 10 v1607 library of IIS10.0 to provide secured communication (TLS1.2) with ZombieZERO Agent and administrator PC.
VMware VIX API 1.14.0	VMware VIX API 1.14.0 is an open application programming interface (API) provided by VMware. It is an API used for ZombieZERO Detector to run the Analyzer Agent for communication with VMware EXSi 5.5 U3.
VMware ESXi 5.5 U3	VMware ESXi 5.5 U3 provides a virtual environment for Analyzer Agent to collect a series of process action information securely by executing files that require malicious code analysis and initialize the execution environment.

.net framework 4.6	.net framework 4.6 is a library to execute application programs developed in the Windows OS product family from Microsoft. It is needed to execute Analyzer Agent.
Microsoft Office 2013	Microsoft Office 2013 is needed for Analyzer Agent to execute Microsoft Office format files to analyze Microsoft Office-related malicious code.
Adobe Acrobat pro XI	Adobe Acrobat pro XI is needed for Analyzer Agent to execute PDF format files to analyze PDF-related malicious code.
Adobe Flash Player 16	Adobe Flash Player 16 is needed for Analyzer Agent to execute SWF format files to analyze SWF-related malicious code.
Hancom Office 2010	Hancom Office 2010 is needed for Analyzer Agent to execute HWP format files to analyze HWP-related malicious code.

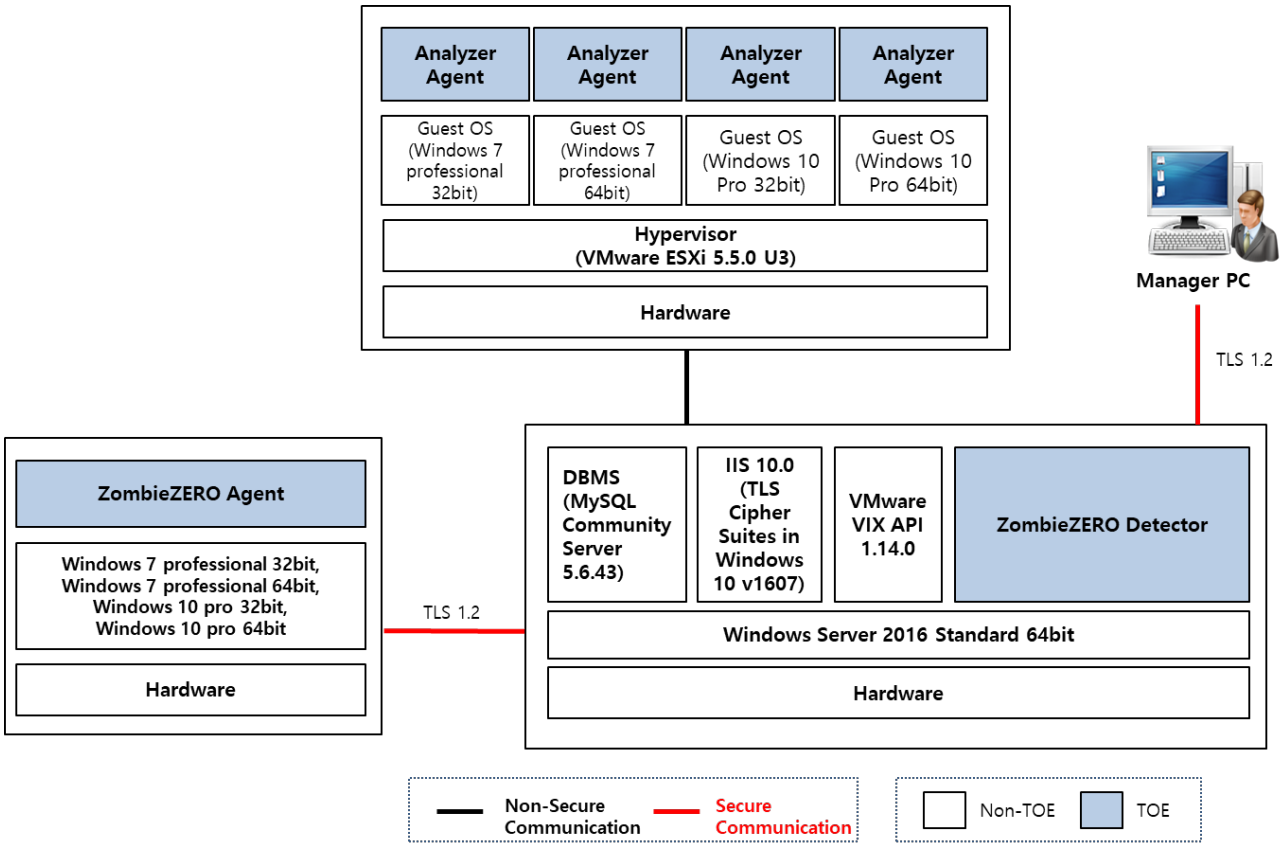
The following external IT entities required to operate the TOE are out of the scope of the TOE

Category	Description
Network Tap	Network Tap is a device that copies traffic from the Internet to the ZombieZERO Detector.
SMS Server	The short message service (SMS) server is located in the external network to send security alarms generated in the ZombieZERO Detector as a form of SMS to the authorized administrator.
Update Server	The update server located in NPCore Inc. provides the up-to-date malicious code information owned by the developer. The updated files are transfer to ZombieZERO Detector using the safe communication channel provided by the operating environment (TLS 1.2 based HTTPS).
Mail Server	The mail server is located in the external network to send security alarms generated in the ZombieZERO Detector as a form of email to the authorized administrator.
Bitdefender Server	Bitdefender server run by Bitdefender Company provides updates to the latest Bitdenfender pattern to ZombieZERO Detector.

1.4 TOE Description

The physical and logical scopes of the TOE are described in the next section.

1.4.1 Physical Scope of TOE



[Figure 2] Physical Scope of TOE

Physical scope of the TOE includes ZombieZERO Detector, Analyzer Agent, and ZombieZERO Agent, which are the components of the TOE as shown in Figure 2. It is provided as a form of digital versatile disc (DVD). In addition, the physical scope includes the “Installation Manual” and “Manual for Administrator” distributed as a form of electronic document (DVD) to the end users (customers).

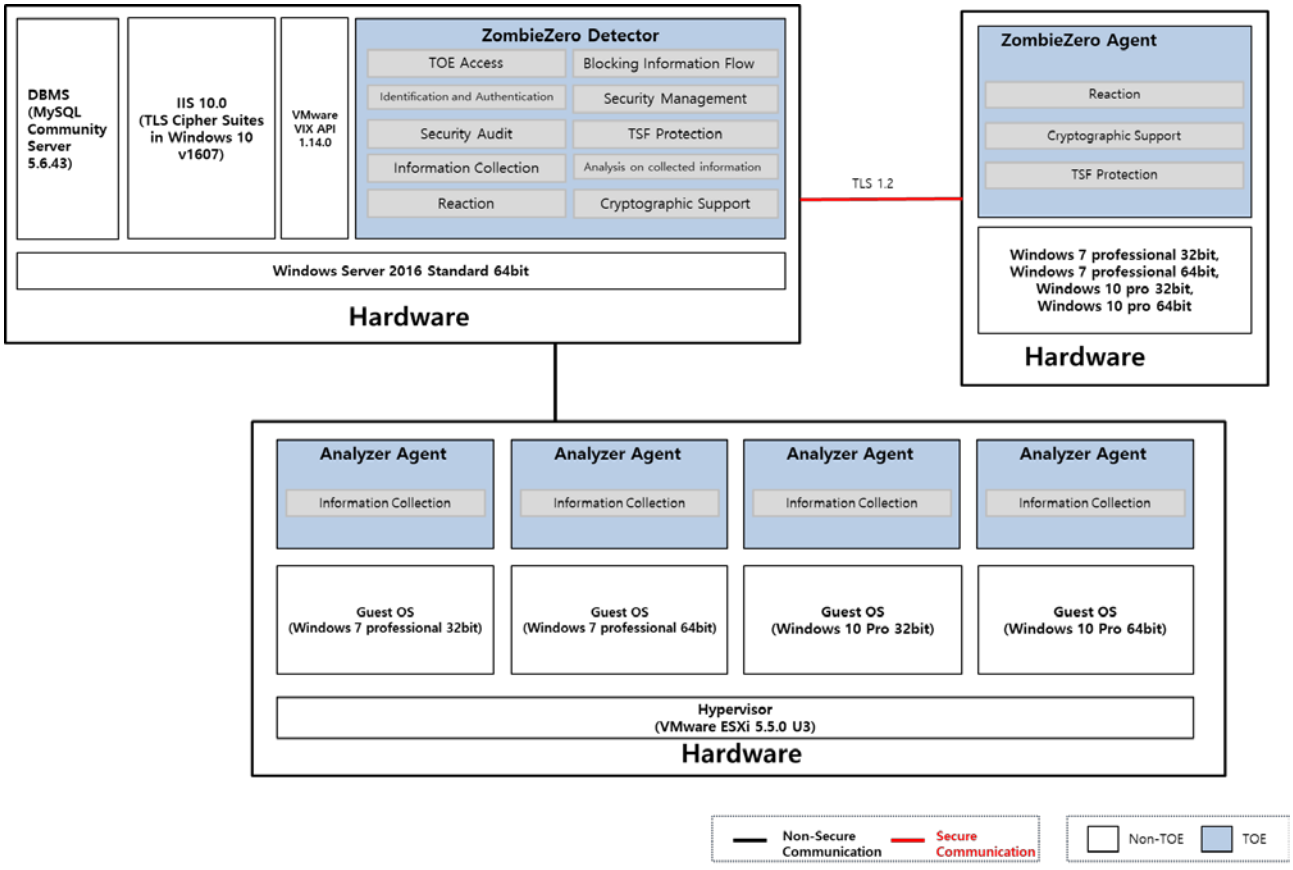
The OS, guest OS, Hypervisor, DBMS, IIS, and VMware VIX API, which are required to operate the TOE, are out of the scope of the TOE

Classification	Distribution form	Type	Distribution
TOE	ZombieZERO Inspector V4.0 (details : V4.0 Revision 12)	-	-

TOE Components	ZombieZERO Detector	ZombieZERO Detector V4.2.71 (ZombieZERO Detector V4.2.71.exe)	Software	DVD
	Analyzer Agent	Analyzer Agent V4.0.0 (Analyzer Agent V4.0.0.exe)		
	ZombieZERO Agent	ZombieZERO Agent V4.0.227 (ZombieZERO Agent V4.0.227.exe)		
Guidance Documents		ZombieZERO Inspector V4.0 Installation Manual V1.4 (ZombieZERO Inspector V4.0 Installation Manual V1.4.pdf)	Electronics File	DVD
		ZombieZERO Inspector V4.0 Manual for Administrator V1.4 (ZombieZERO Inspector V4.0 Manual for Administrator V1.4.pdf)		

1.4.2 Logical Scope of TOE

Logical scope of the TOE is shown in the following [figure 3]



[Figure 3] Logical Scope of TOE

The major security features of the TOE are as follows

Component	Description of the security features
ZombieZERO Detector	▪ Security Audit - Audit data such as log data of file detection, malicious code detection, and file analysis results, and administrator login and setup change are generated. - It provides a function that queries audit records for authorized administrators. - It sends email and SMS to the authorized administrator if the audit data are predicted to be lost, and overwrites audit log records, which are outdated if the audit trail data are full.
	▪ Cryptographic Support - Functions of encryption (AES 128) of policy files and signature verification (RSA 2048) of update files are provided. - Functions of hash generation (SHA 384) of administrator password and hash generation (SHA 384) of execution code for integrity inspection are provided. - Functions of encryption key generation and public key destruction (overwrite with random numbers)
	▪ Identification and Authentication - ID/password-based identification and authentication function is provided during the security management connection. - The mandatory password combination rule (at least nine characters up to 15 characters using alphabet upper case, lower case, numbers and special characters) shall be enforced and authentication feedback (“*” is displayed when password is entered) protection function is provided. - Account lock function is provided when authentication fails five times consecutively (in case of administrator privilege users: 10 min lock, others: locked until top administrator activates the account).
	▪ Security Management - A function that can perform security management (administrator setup, query of audit record, agent management, and inspector management etc.), set up and manage TSF data by authorized administrators (top, execution, read-only, and Monitoring administrator) according to the role of administrator.
	▪ TSF Protection - Data integrity of ZombieZERO Detector installation files is inspected at the time of startup, during operations periodically (one, three, six, and 12-hour) and upon the request from the authorized administrator.
	▪ TOE Access - Administrator session is terminated after some inactivity time (5-10 min, default value is 10 min.) of the authorized administrator.

	▪ Blocking Information Flow - Information flow is blocked by sending a reset packet to the target user PC when a packet is Transmission from an internal user PC to the IP/URL addresses registered in the list of information flow block.
	▪ Information Collection - Files are collected from the inflow traffic to the internal networks through HTTP/1.1, FTP, POP3, IMAP, and SMTP protocols. - Action information of files collected by Analyzer Agent is stored. - Files whose file extension is ".exe", which are requested to be analyzed by ZombieZERO Agent, are stored.
	▪ Analysis on collected information - Whether the collected analysis target files contain malicious code is detected by comparing them with the signature pattern. - Whether the collected analysis target files contain malicious code is detected by comparing them with the detection rules. - Whether the action information of the files contains malicious code is detected by comparing them with the detection rules.
	▪ Reaction - Signature is generated against the detected malicious code. - The block/isolation policy against the detected malicious code is transferred to ZombieZERO Agent.
Analyzer Agent	▪ Information Collection - Execution information about registry action, network action, memory action, file action, and process action is collected and stored by executing the files transferred from ZombieZERO Detector.
ZombieZERO Agent	▪ Reaction - Execution of malicious code files is blocked or isolated according to the malicious code block and isolation policy.
	▪ Cryptographic Support - Decryption of security policy file is provided. - Creating a hash of registry value (SHA 384) of ZombieZERO Agent is provided. - Function to destroy encryption keys (overwrite with zeroization '0') is provided.
	▪ TSF Protection - Data integrity of registry setup information is inspected upon the request from the authorized administrator at the time of startup, and during operations periodically (one, three, six, and 12-hour).

1.5 Convention

This Security Target mixed uses some abbreviation for clear understanding. Notations, forms, and conventions used in this document follow the Common Criteria (Notification No. 2013-51 by the Ministry of Science, ICT and Future Planning) (hereafter referred to as CC).
The CC allow selection, assignment, refinement, and iteration operations that can be performed in Security Functional Requirements. Each operation is used in this Security Target, and denoted as follows.

Iteration

The iteration operation is used when an operation is multiply applied, and one component is repeated many times. The result of iteration operation is marked as an iteration number in parenthesis after component identifier, i.e., (Iteration No.).

Assignment

The assignment operation is used for parameter assignment that is not specified (e.g., password length). The result of Assignment operation is marked as squared parenthesis, i.e., [assignment_value].

Selection

The selection operation is used when more than one selectable option among Information Protection System Common Criteria while requirement writing. The result of Selection operation is marked as *underlined italicized text*.

Refinement

The refinement operation is used for requirement limitation by adding specification on requirements. The result of Refinement operation is marked as **bold text**.

1.6 Terms and Definitions

Top Administrator

Top Administrator is an authorized administrator with full privileges. Top Administrator can use all the functions specified in security management, add / delete administrators, and assign / change roles.

Operation Administrator

Operation administrator is an authorized administrator with full privileges. Operation administrator can use all the functions specified in security management, add / delete administrators except top privilege, and assign / change roles.

Read only Administrator

Read only Administrator is an authorized administrator who can read the information in the TOE. Read-only administrator can access a monitoring menu and search audit records.

Monitoring Administrator

Monitoring Administrator is an authorized administrator who has a privilege to check only monitoring menu contents in the TOE.

Malicious Code [Source: Korean Telecommunications Technology Association (TTA) Glossary]

It refers to executable code created by malicious purposes. Executable code includes not only program, macro, and script but also data format using vulnerability. Malicious software is the widest concept. It can be classified to viruses, worms, Trojan horses, and spyware according to the self-replication ability and infection target.

* Term Source: Glossary from the Telecommunications Technology Association (TTA).

Malicious behavior

Action that uses user's PC in malicious activities as malicious code is installed in user's PC with the malicious purpose.

Behavior-based

It is a technology that detect malicious code by determining whether the code is malicious based on the behavior of the executable file to respond to the malicious code effectively which is advanced daily and difficult to be detected by existing pattern methods.

Network TAP

Network TAP is an external monitoring device that mirrors traffic delivered between network nodes. It is a hardware device inserted to the specific spot (intrusion prevention system (IPS) and switch to monitor data.

Port Mirroring

It is a method to replicate packets to other switch ports to monitor or observe the packets through the network switch.

Malicious File

Files that contain malicious code.

Blacklist

A list of malicious files and malicious code distribution sites (IP/URL) held by the developer and files that are determined as malicious files by the ZombieZERO Detector.

Whitelist

A list of files that are determined as normal files by the ZombieZERO Detector.

2. Conformance Claims

2.1 Conformance to Common Criteria

This Security Target conforms to the following Common Criteria :

o Common Criteria Identification

- Information Protection System Common Criteria, Part 1: Introduction and General Model, April 2017, Version 3.1 Revision 5, CCMB-2017-04-001
- Information Protection System Common Criteria, Part 2: Security Functional Requirements, April 2017, Version 3.1 Revision 5, CCMB-2017-04-002
- Information Protection System Common Criteria, Part 3: Assurance Components, April 2017, Version 3.1 Revision 5, CCMB-2017-04-003

o Common Criteria Conformance

- Extension of Information Protection System Common Criteria, Part 2.
  - DSM_COL.1 file collection
  - DSM_COL.2 file execution information collection
  - DSM_INA.1 information analysis
  - DSM_RCT.1 Security Reaction
- Conformance of Information Protection System Common Criteria, Part 3.

2.2 Conformance to Protection Profile

There is no Protection Profile that his Security Target conforms.

2.3 Conformance to Package

This Security Target is in conformance with the Assurance Package EAL 2.

2.4 Conformance Claim Rationale

Since this Security Target does not claim the Protection Profile conformance, it does not need the conformance claim rationale.

3. Security Problem Definition

The security problem definition defines intended threats, organizational security policies, and assumptions to discuss the TOE and TOE operational environment.

The primary assets that TOE protects are defined as major system files and TSF data, which make up the systems in the user's PC in the internal network, and TOE and TSF data that protect the major system files and data.

3.1 Threats

Generally, threat agents are IT entities and users that pose a threat by distributing malicious code via unauthorized access or abnormal means to the assets that are needed be protected by the TOE. They may generate various threats as follows:

T. Recording Failure

Threat agent may have the audit data not to be stored by exhausting the storage capacity.

T. Masquerade

Threat agents may access the TOE by masquerading as the authorized administrator.

T. Consecutive authentication attempts

Threat agents may acquire authority of authorized user by consecutive authentication attempts for TOE access.

T. Re-use Attack

Threat agents may access the TOE security management by reusing administrator authentication data.

T. Internal Network Invasion via Malicious Code

Threat agents may invade the resources in the internal network or leak internal information to the outside through malicious code.

T. Transmission Data Damage

Threat agents may illegally expose or change the data that transmitted between TOE components or between TOE and update server by unauthorized manner.

T. Stored Data Damage

Threat agents may illegally expose or change the data stored inside the TOE by unauthorized manner.

T. Malicious Code Propagation during Action Analysis

The threat agent may infect or propagate malicious code to the TOE or the network connected with the TOE when the TOE executes the target file for malicious code detection.

T. Unauthorized File Execution

Threat agents may bypass the security functions of the TOE by installing or executing unauthorized files that violate the security policies in the internal user PCs.

T. Access to Unauthorized External Network

When an internal user PC accesses to an unauthorized external server (e.g., malicious bot distribution server, C&C server) that violates the security policy, it may be infected with a malicious code or an unauthorized command may be executed from it.

T. TSF Damage

Threat agents may induce malfunctions of TOE functions or disable TOE functions by damaging TSF through unauthorized accesses.

3.2 Organizational Security Policies

P. Audit

The security-related incidents shall be recorded and archived to trace the responsibility of all actions in relation to security, and the recorded data shall be investigated. In addition, available spaces of the disks for audit data storage shall be inspected on a regular basis to prevent the loss of audit data and unauthorized modification and deletion to the stored audit data shall not occur.

P. Secured Operation

Administrator shall set up a TOE securely to comply with the organizational security policy and provide a management mean to operate the TOE accurately according to the TOE Operation Manual.

P. Cryptographic Strength

Organizations shall apply an encryption measure to confidential data storage such as administrators' password and their transmission routes, and employ a secured encryption algorithm.

3.3 Assumptions

A. Physical Security

ZombieZERO Detector and Analyzer Agent among the TOE components shall be located in a physically secured environment where only authorized administrators can access.

A. Security Maintenance

When the internal network environments change due to the modification of network configuration, increase or decrease in the number of hosts, and service increase or decrease, the security shall be maintained as the same level as the previous one by reflecting the changed environments and security policies immediately to the TOE operation policies.

A. Trusted Administrator

Authorized administrator of TOE shall be non-malicious, have completed appropriate training on TOE administration functions and fulfill obligations according to administrator guidelines.

A. Operating System Reinforcement

Reliability and security of an operating system shall be ensured by administering operations to remove services or means in operating system not required and reinforcement on vulnerabilities in the operating system.

A. DBMS

DBMS that stores the audit and TSF data shall be operated in a physically secured environment.

A. Timestamp

The TOE operation environment shall provide a reliable timestamp.

A. Virtual Machine

Among the TOE components, the Analyzer Agent runs and operates securely on virtual machines provided by Hypervisor.

A. TOE Environment Configuration

The internal and external communications shall be done only through the network TAP to enable the TOE to perform mirroring of network traffic that is introduced to the internal network or Transmission to the Internet. In addition, Analyzer Agent is directly interlinked with ZombieZERO Detector without a network route.

4. Security Objectives

This chapter describes the security objectives for the TOE and the security objectives for the TOE environment.

4.1 Security Objectives for the TOE

O. Audit

The TOE shall record and maintain security-related incidents in order to enable tracing of responsibility for security-related acts and must provide means to review the recorded data. In addition, it shall provide a countermeasure feature to maintain the audit function when the storage space of audit data exceeds the threshold and reach the full storage state.

O. Security Management

The TOE shall provide a secured means to manage TSF and TSF data efficiently, and only authorized administrators shall perform such important management functions and data management. Moreover, the update shall be applied securely after signature verification on the update file transmitted from the update server.

O. Identification and Authentication

The TOE shall solely identify and securely authenticate administrators who like to connect to the TOE and perform authorized administrator's role.

O. Secure Cryptographic

The TOE shall use encryption algorithms with secured security strength for encryption, electronic signature verification, and hash operation, and encryption key generation and destruction used in encryption operation shall be securely performed.

O. Stored Data Protection

The TOE shall protect stored TSF data from unauthorized exposure and change.

O. Detection and Reaction to Malicious Code

The TOE shall detect malicious code and perform the reaction (block or isolation) through information collection and analysis to detect malicious code introduced through the network.

O. Unauthorized Information Flow Blocking

The TOE shall block the information flow to the unauthorized external servers (e.g., malicious bot distribution server, C&C server) at the internal users' PCs.

O. Self-protection

The TOE shall test whether the major hardware and TSF (main processes in the TOE security functions) are run normally during the execution, and authorized administrators shall be able to verify the test results. In addition, the integrity of the main processes in the TOE security functions shall be verified.

4.2 Security Objectives for the Operational Environment

OE. Physical Security

The TOE (ZombieZERO Detector and Analyzer Agent) shall be located in a physically secured environment where only authorized administrators can access.

OE. Security Maintenance

When the internal network environments change due to the modification of network configuration or increase or decrease in the number of internal user PCs, the security shall be maintained as the same level as the previous one by reflecting the changed environments and security policies immediately to the TOE operation policies.

OE. Trusted Administrator

Authorized administrator of TOE shall be non-malicious, have completed appropriate training on TOE administration functions and fulfill obligations according to administrator guidelines.

OE. Operating System Reinforcement

Reliability and security of operating system must be assured by administering operations to remove all unnecessary services or means in operating systems and perform reinforcement on vulnerabilities in the system.

OE. DBMS

DBMS that stores the audit and TSF data shall be operated in a physically secured environment.

OE. Timestamp

The TOE shall record security-related incidents accurately by using a reliable timestamp that is provided by the TOE operational environment.

OE. Transmission Data Protection

The TOE operational environment shall provide a security management GUI through TLS-based web browsers and ensure the reliability and security by using the TLS protocol communication mechanism provided by the TOE operational environment when providing a communication channel for secured communication between TOE components (ZombieZERO Detector and ZombieZERO Agent).

OE. Virtual Machine

Among the TOE components, the Analyzer Agent must be securely executed and operated in the virtual machine provided by the Hypervisor.

OE. TOE Environment Configuration

The communication between the Internet and internal network shall be done only through the network TAP, and packets mirrored through the network TAP shall be Transmission to the TOE (ZombieZERO Detector). In addition, Analyzer Agent shall be directly interlinked with ZombieZERO Detector without a network route.

OE. Secure Update

The update server shall transmit the electronic signed (RSA2048) update files to the TOE using a secure communication channel (TLS 1.2-based HTTPS) provided by the operating environment.

4.3 Security Objective Rationale

The Rationale of Security Objective demonstrates that the specified security objectives are appropriate, sufficient to handle security problems and are essential, rather than excessive. The Rationale of Security objective demonstrates the following items. Each assumption, threat, and security policy of the organization is handled by at least one security objective. Each security objective handles at least one assumption, threat, and security policy of organization.

[Table 3] Security Objective Rationale

	O.Audit	O.Audit management	O.Identification and Authentication	O. Secure Cryptographic	O. Stored Data Protection	O.Detection and reaction to malicious code	O.Blocking unauthorized information flow	O.Self-protection	OE.Physical Security	OE.Security Maintenance	OE.Trusted Administrator	OE.Operating System Reinforcement	OE.DBMS	OE.Timestamp	OE.Transmission Data Protection	OE. Virtual Machine	OE. TOE Environment Configuration	OE. Secure Update
T.Recording Failure	✓																	
T.Masquerade			✓															
T.Continuous Authentication Attempt			✓															
T.Re-use Attack			✓															
T.Internal Network Invasion via Malicious Code						✓												
T.Transmission Data Damage															✓			✓
T.Stored Data Damage				✓														
T.Malicious Code Propagation during Action Analysis																✓		
T.Unauthorized File Execution						✓												

	O.Audit	O.Audit management	O.Identification and Authentication	O. Secure Cryptographic	O. Stored Data Protection	O.Detection and reaction to malicious code	O.Blocking unauthorized information flow	O.Self-protection	OE.Physical Security	OE.Security Maintenance	OE.Trusted Administrator	OE.Operating System Reinforcement	OE.DBMS	OE.Timestamp	OE.Transmission Data Protection	OE. Virtual Machine	OE. TOE Environment Configuration	OE. Secure Update
T.Access to Unauthorized External Network							✓											
T.TSF Damage				✓				✓										
P.Audit	✓													✓				
P.Secured Operation		✓									✓							
P. Cryptographic Strength		✓		✓	✓													✓
A.Physical Security									✓									
A.Security Maintenance										✓								
A.Trusted Administrator											✓							
A.Operating System Reinforcement												✓						
A.DBMS													✓					
A.Timestamp														✓				
A.Virtual Machine																✓		
A.TOE Environment Configuration																	✓	

4.3.1 Security Objectives Rationale for TOE

O. Audit

The TOE ensures providing a means to record, maintain and investigate security-related incidents accurately in detail. Thus, this security objective of the TOE is needed to respond to Threat ‘T.Recording Failure’ and perform Security Policy ‘P.Audit’.

O. Security Management

The TOE provides authorized administrators with a means to manage the TOE securely and ensures the update file application after signature verification of the update files received from the update server. Thus, it is required to perform Organizational Security Policy ‘P.Secured Management’ and ‘P.Password Strength’.

O. Identification and Authentication

The TOE ensures identification and authentication of the users uniquely. Thus, this TOE security objective is needed to respond to Threats: ‘T.Masquerade’, ‘T.Continuous Authentication Attempt’, and ‘T.Re-use Attack’.

O. Secure Cryptographic

The TOE employs the encryption algorithms (ASE128, RSA2048, SHA384) which have a secure security strength for encryption, electronic signature verification, and hash operation, and performs a generation (AES 128) and destruction (overwrite with random numbers) of encryption keys used in encryption operation securely. Thus, the security objective of the TOE is required to cope with Threat ‘T.Stored Data Damage’ and ‘T.TSF Damage’, and perform Organizational Security Policy ‘P.Password Strength’.

O. Stored Data Protection

The TOE assures the stored data protection by storing policy files encrypted with ASE (128-bit) encryption algorithm to protect its own TSF data. Thus, it is required to perform Organizational Security Policy ‘P.Password Strength’.

O. Detection and Reaction to Malicious Code

Attackers may distribute malicious code in the Internet and have the internal network computers to be infected by malicious code. To prevent the internal network from being infected with malicious code, the TOE collects the files from the traffic coming from the external network and collects the files (.exe) transmitted from the internal network computer during TOE operation. The TOE uses the Bitdefender pattern and Yara rules to detect malicious code. Depending on the detection result, the TOE blocks or isolates the detected malicious

code to ensure malicious code detection and response. Thus, this objective is needed to respond to Threats : ‘T.Internal Network Invasion via Malicious’ and ‘T.Unauthorized File Execution’.

O. Unauthorized Information Flow Control

The users in the internal network may download files that contain malicious code after they connect to a malicious code distribution site (IP/URL). To cope with this circumstance, the TOE ensures blocking the access to the unauthorized external network by sending a reset packet to the computers in the internal network based on the list of malicious code distribution sites when the computers in the internal network attempt a connection to one of the malicious code distribution sites (IP/URL). Thus, it supports Threat ‘T.Access to Unauthorized External Network’.

O. Self-protection

The TOE ensures to verify whether the TOE is modulated by performing a periodical test at the initial execution or during operation to see the correction operation of the TOE, and sending an alarm message via SMS or email of the authorized administrator if the test fails. Thus, this TOE security objective is needed to respond to Threat ‘T.TSF Damage’.

4.3.2 Security Objectives Rationale for Operational Environment

OE. Physical Security

This security objective is required to support assumption of ‘A.Physical Security’ since the TOE guarantees physical security.

OE. Security Maintenance

This security objective guarantees that the TOE reflects modified environment and security policy on TOE operation policy immediately when internal network environment is modified such as internal network configuration changing, Host increase/decrease, or Service increase/decrease so that it maintains the same security level with the prior one. Thus, it supports Security Policy ‘A.Security Maintenance’.

OE. Trusted Administrator

This security objective is required to perform Organizational Security Policy ‘P.Secured Operation’ and support Assumption ‘A.Trusted Administrator’ since it guarantees that authorized administrator of TOE can be reliable.

OE. Operating System Reinforcement

This security objective is required to support assumption of ‘A.Operating System Reinforcement’ since it guarantees that the Operation System is safe and reliable by eliminating all the unnecessary services or means on the OS and operating reinforcement task on OS vulnerability.

OE. DBMS

This security objective ensures that TSF data (audit records, security policy, and environment setup) provided by the TOE are stored in DBMS and the DBMS can only be accessed through the TOE. In addition, this security objective is required to support Assumption ‘A.DBMS’ since it guarantees that authorized administrator can ensure the normal audit record generation by managing an available space of DBMS storage.

OE. Timestamp

This security objective is required to perform Organizational Security Policy ‘P.Audit’ and support Assumption ‘A.Timestamp’ since the TOE ensures accurate recording of security-related incidents by using a reliable timestamp provided by the TOE's operational environment.

OE. Transmission Data Protection

This security objective is required to prevent Threat ‘T.Transmission Data Damage’ since the TOE protects Transmission data from being tempered between the TOE components by employing HTTPS (TLS 1.2) protocol provided by IIS 10.0.

OE. Virtual Machine

To protect against malicious code infection and external spread, the TOE's malicious code analysis environment uses a virtual machine, which is separated from the external network when suspected malicious files are executed. In addition, after analyzing the malicious code, it supports the safe analysis environment by using the virtual machine initialization function provided by Hypervisor. It is needed to support ‘A.Virtual machines’.

OE. TOE Environment Configuration

The operational environment of the TOE ensures the use of auxiliary system (network TAP) and installation configuration environment to enable the detection of all traffic between users and external networks. Thus, this security objective supports Security Policy ‘A.TOE Environment Configuration’.

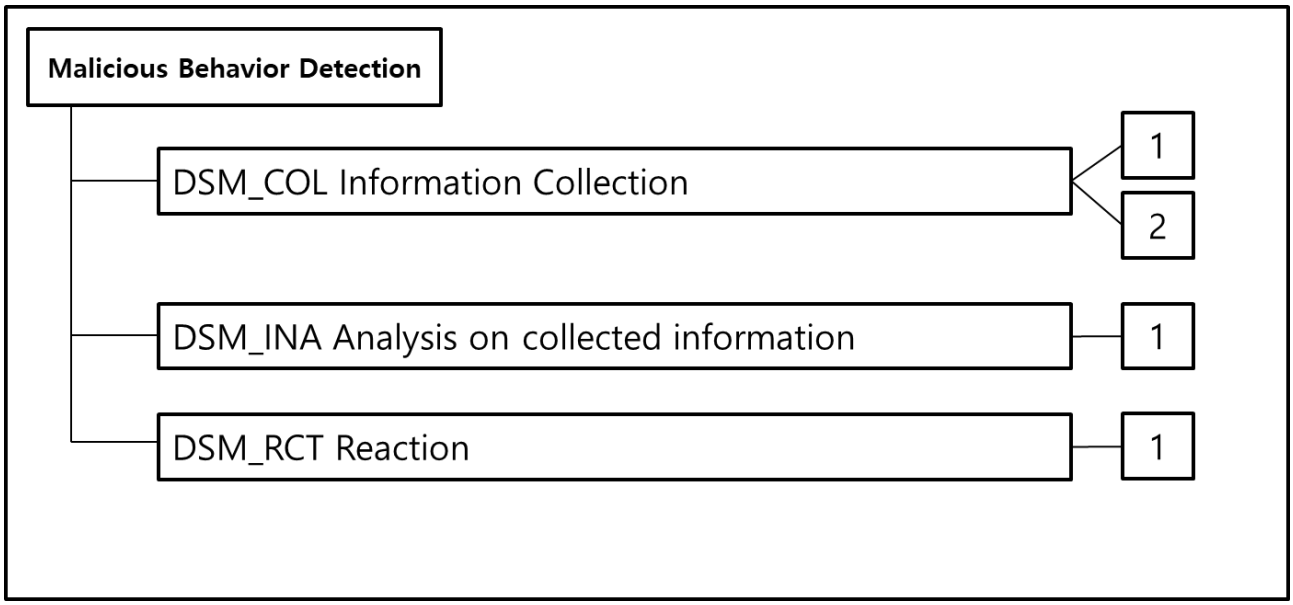
OE. Secure Update

The update server ensures that the electronic signed (RSA2048) update files are sent to the TOE using a secure communication channel (TLS 1.2-based HTTPS) provided by the operating environment. Thus, the security objective of the TOE is required to cope with Threat ‘T.Transmitted Data Damage’ and to perform Organizational Security Policy ‘P.Password Strength’.

5. Extended Components Definition

5.1 Class DSM : Malicious Behavior Detection

The detection system for malicious behavior (DSM) defines the requirements about information collection to detect malicious behavior and malicious code that may induce malicious behavior, analysis of the collected information, and reactions.



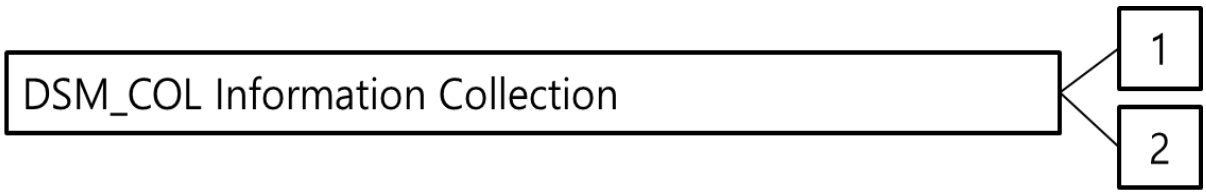
[Figure 4] Configuration of the Malicious Behavior Detection Class

5.1.1 Information Collection (DSM_COL)

Family Behaviour

The information collection (DSM_COL, Collection) family defines the requirements that provide information collection capabilities to analyze malicious code and behaviors.

Component levelling



DSM_COL.1 File collection defines a file type to be collected and designates how to collect.

DSM_COL.2 The collection of file behavior's information specifies the file type to be executed and the method of storing the behavior history.

Management: DSM_COL.1
The flowing actions could be considered for the management functions in FMT:

a) Management of security attributes regarding collection targets.

Management: DSM_COL.2
There are no management activities foreseen.

Audit: DSM_COL.1, DSM_COL.2
There are no auditable events foreseen.

DSM_COL.1 file collection
Hierarchical to: No other components.
Dependencies: None
DSM_COL.1.1 TSF shall extract and store files [assignment: *collection-required file extensions*] over the protocols [Assignment: *analysis-required protocols*] for malicious code detection.
DSM_COL.1.2 TSF shall generate information [assignment: *collected file information*] about collected files.

DSM_COL.2 file execution information collection
Hierarchical to: No other components.
Dependencies: No dependencies
DSM_COL.2.1 TSF shall execute files of the file extensions [assignment: *file extensions that can be executable and collect information*] for malicious behavior detection.
DSM_COL.1.2 TSF shall monitor executed files [assignment: *operation history of monitoring-required files*] and store the behavior record in a file format.

5.1.2 Information Analysis (DSM_INA)

Family Behaviour
The information analysis (DSM_INA, INformation Analysis) family defines the requirements that detect malicious behaviors or malicious code that induces malicious behaviors through the analysis on the collected data by DSM_COL.

Component levelling

DSM_INA.1 Information Analysis is required to perform analyses based on collected data, and generate related information after detecting malicious behaviors or malicious code that induces malicious behaviors among data generated while the analysis is performed.

Management: DSM_INA.1
The flowing actions could be considered for the management functions in FMT:
a) Retaining the list of malicious code/malicious behavior detection (addition, deletion)

Audit: DSM_INA.1
The flowing actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) Minimal: Operation initiation and suspension of the analysis mechanism.

DSM_INA.1 information analysis
Hierarchical to: No other components
Dependencies: None
DSM_INA.1.1 TSF shall perform the [assignment: *analysis function*] based on the collected data.
DSM_INA.1.2 TSF shall generate the following information after analyzing the collected data.
a) Analysis end date, analysis-required file name and extension, file size, and analysis result (malicious or not)
b) [assignment: *other information*]

5.1.3 Reaction (DSM_RCT)

Family Behaviour
The Reaction (DSM_RCT, Reaction) family defines malicious code signature generation that shall be performed when malicious behaviors or malicious code that induces malicious behaviors are detected through the execution of unknown files, and malicious code block or isolation actions according to the reaction policy transmission and reaction policy reception regarding malicious code block or isolation.

Component levelling

DSM_RCT.1 Security Reaction shall take a reaction if security violation is likely to occur or violation incidents are detected.

Management: DSM_RCT.1
The flowing actions could be considered for the management functions in FMT:
a) Management of reactions (No reaction~~,~~ block, isolation)

Audit: DSM_RCT.1
The flowing actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) Minimal: reactions taken due to the malicious code/behavior detection.

DSM_RCT.1 reaction guarantee
Hierarchical to: No other components
Dependencies: DSM_INA.1 information analysis

DSM_RCT.1.1 TSF shall perform [assignment: reactions] if malicious behavior/malicious code is detected [selection: detection and reaction policy reception, [assignment: method to acquire malicious behavior/malicious code-related information]].

6. Security Requirements

The security requirements define security functional requirements that satisfy the TOE security objective and TOE security assurance components to obtain the assurance that the TOE satisfies the security functional requirements.

All the subjects/objects, operations, and security attributes use in the security requirements in the ST are defined as presented in the following table.

Subject	Security attribute of the subject	Object	Security attribute of the object	Operation
User PC installed TOE	▪ IP address	Traffic Transmitting from user PC to external network	▪ Destination IP ▪ Destination URL	▪ N/A ▪ Detection ▪ Block
User PC installed TOE	▪ IP address	Files in traffic introduced from external to internal networks by the user request	▪ File name (extension)	▪ Collection/storage ▪ Analysis
		Files Transmittied from user PC	▪ exe files	▪ Collection/storage ▪ Analysis
User PC installed TOE.	▪ IP address	Files executed in user PC	▪ exe files	▪ Transmission ▪ Detection ▪ Block ▪ Isolation

6.1 Security Functional Requirements

This Security Target includes the security functional components defined in the below table from the CC part 2 and Extended Component Definition as the security functional requirements.

[Table 4] Summary of Security Functional Requirements

Components	Security Functional Groups	Security Functional Requirements	
	Security Audit	FAU_GEN.1	Audit Data Generation
		FAU_GEN.2	User Identity Association

Components	Security Functional Groups	Security Functional Requirements	
ZombieZERO Detector		FAU_SAR.1	Audit Review
		FAU_SAR.3	Selectable Audit Review
		FAU_STG.1	Protected Audit Trail Storage
		FAU_STG.3	Action in case of Possible Audit Data loss
		FAU_STG.4	Prevention of Audit Data Loss
	Cryptographic Support	FCS_CKM.1	Cryptographic key generation
		FCS_CKM.4(1)	Cryptographic key destruction
		FCS_COP.1(1)	Cryptographic operation
		FCS_COP.1(2)	Cryptographic operation
		FCS_COP.1(3)	Cryptographic operation
	User Data Protection	FDP_IFC.1	Subset Information Flow Control
		FDP_IFF.1	Simple Security Attributes
	Identification and Authentication	FIA_AFL.1	Authentication Failure Handling
		FIA_ATD.1	User Attribute Definition
		FIA_SOS.1	Verification of Confidential Information
		FIA_UAU.2	User Authentication Before Any Action
		FIA_UAU.4	Single-use Authentication Mechanisms
		FIA_UAU.7	Protected Authentication Feedback
		FIA_UID.2	User Identification Before Any Action
	Security Management	FMT_MOF.1	Management of Security Functions Behavior
		FMT_MSA.1	Management of Security Attributes
		FMT_MSA.3	Static Attribute Initialization
		FMT_MTD.1	Management of TSF Data
		FMT_SMF.1	Specification of Management Functions
		FMT_SMR.1	Security Roles
	TSF Protection	FPT_TST.1(1)	TSF Testing
	TOE Access	FTA_SSL.3	TSF-initiated Termination
	Information Collection	DSM_COL.1	File Collection

Components	Security Functional Groups	Security Functional Requirements	
	Analysis on Collected Information	DSM_INA.1	Information Analysis
	Reaction	DSM_RCT.1(1)	Security Reaction
Analyzer Agent	File Execution	DSM_COL.2	File Execution Information Collection
ZombieZERO Agent	Reaction	DSM_RCT.1(2)	Security Reaction
	Cryptographic Support	FCS_CKM.4(2)	Cryptographic key destruction
		FCS_COP.1(4)	Cryptographic operation
		FCS_COP.1(5)	Cryptographic operation
	TSF Protection	FPT_TST.1(2)	TSF Testing

6.1.1 ZombieZERO Detector

6.1.1.1 Security Audit

FAU_GEN.1 Audit Data Generation

Hierarchical to: No other components.
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) [The events listed in [Table 5]]

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].

[Table 5] Auditable Events

Requirement	Auditable events	Additional Audit Record Contents

FAU_SAR.3	None.	None.
FAU_STG.3	Action in case of threshold exceed	None.
FAU_STG.4	Action in case of audit storage failure	None.
FIA_AFL.1	Action when the number of failed authentication attempts exceed the threshold and taken reaction	None.
FIA_SOS.1	Denial of all tested secrete information by TSF	None.
FIA_UAU.2	Use of all authentication mechanisms	None.
FIA_UAU.4	Attempts of authentication data reuse	None.
FIA_UID.2	Failure of user identification mechanism including the provided user identity	None.
FMT_MOF.1	All changes to TSF functions	None.
FMT_MTD.1	All management activities of TSF data	Modified values of TSF data
FMR_SMF.1	Use of management functions	None.
FMT_SMR.1	Changes to user groups to divide a role	None.
FPT_TST.1	TSF Testing execution and test results	None.
FTA_SSL.3	Termination of interaction session due to session lock mechanism	None.
DSM_INA.1	Operation initiation and suspension of analysis mechanism Detailed detection results of malicious code/behavior	None.

FAU_GEN.2 User Identity Association

Hierarchical to: No other components.
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.

FAU_SAR.1 Audit Review

Hierarchical to: No other components.
Dependencies: FAU_GEN.1 Audit Data Generation

FAU_SAR.1.1 The TSF shall provide [top administrator, operation administrator, read-only administrator] with the capability to read [agent logs, IP/URL Block Log, general logs, analysis event logs, and analysis result logs] from the audit records.

[Table 6] Details of Provided Audit Review by Administrator

[O: read-only right provided, X: read-only right not provided]

Audit record type	Top Administrator	Operation administrator	Read-only administrator	Monitoring administrator
Agent log	O	O	O	X
IP/URL Block Log	O	O	O	X
General log	O	O	O	X
Analysis event log	O	O	O	X
Analysis result log	O	O	O	X

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

* Application notes: The audit data generated by audit record (log) type are as follows:
- Agent log: Time information (detection, blocking, or isolated time) when reaction occurs in ZombieZERO Agent, policy (detection, blocking, or isolation) information, and user information (user PC name, MAC address, and IP address) are provided.
- IP/URL Block Log: Detected/isolated time information based on IP/URL defined in unauthorized server list, policy (detection and isolation) information, and destination (IP and URL) information.
- General log: Security policy modification, system setup modification, self-test errors, authorized administrator login or logout, and administrator authentication failure.
- Analysis event log: Start and stop malicious code detection (based on Bitdefender and Yara rules) and Analyzer Agent actions
- Analysis result log: Detection results of malicious code based on collected information and Bitdefender pattern, detection results of malicious code based on collected information and Yara rules, and detection results of malicious code based on file execution information and Yara rules are provided.

FAU_SAR.3 Selectable Audit Review

Hierarchical to: No other components.
Dependencies: FAU_SAR.1 Audit Review

FAU_SAR.3.1 The TSF shall provide the ability to apply [search and sorting] of audit data based on [event date, classification, count, keywords, analysis results, and importance].

FAU_STG.1 Protected Audit Trail Storage

Hierarchical to: No other components.
Dependencies: FAU_GEN.1 Audit Data Generation

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.
FAU_STG.1.2 The TSF shall be able to *prevent* unauthorized modifications to the stored audit records in the audit trail.

FAU_STG.3 Action in case of Possible Audit Data Loss

Hierarchical to: No other components.
Dependencies: FAU_STG.1 Protected Audit Trail Storage

FAU_STG.3.1 The TSF shall [take an action of notification and generation of audit data to top administrator, operation administrator, read-only administrator, and monitoring administrator] if the audit trail exceeds [the designated size (below 5 to 10% of the hard disk available space, default value is 10%].

FAU_STG.4 Prevention of Audit Data Loss

Hierarchical to: FAU_STG.3 Action in case of Possible Audit Data Loss
Dependencies: FAU_STG.1 Protected Audit Trail Storage

FAU_STG.4.1 The TSF shall *overwrite the oldest stored audit records* and [send email and SMS to designated administrators] if the audit trail is full.

6.1.1.2 Cryptographic Support

FCS_CKM.1 Cryptographic key generation

Hierarchical to: No other components.
Dependences: [FCS_CKM.2 Cryptographic 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 [AES CBC Mode] and specified cryptographic key sizes [128-bit] that meets the following: [Section 5.2 in ISO/IEC 18033-3:2010].

FCS_CKM.4(1) Cryptographic key destruction

Hierarchical to: No other components.
Dependences: [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 [overwrite with random numbers] that meets the following: [none].

FCS_COP.1(1) Cryptographic operation

Hierarchical to: No other components.
Dependences: [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 [Operating encryption of malicious code blocking/isolate policy files] in accordance with a specified cryptographic algorithm [AES CBC Mode] and cryptographic key sizes [128-bit] that meets the following: [Section 5.2 in ISO/IEC 18033-3:2010].

FCS_COP.1(2) Cryptographic operation

Hierarchical to: No other components.
Dependences: [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 [update file signature verifications] in accordance with a specified cryptographic algorithm [RSA] and cryptographic key sizes [2048-bit] that meets the following: [ISO/IEC 14888-2(2008)].

FCS_COP.1(3) Cryptographic operation

Hierarchical to: No other components.
Dependences: [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 [ZombieZERO Detector execution code and administrator password hash generations] in accordance with a specified cryptographic algorithm [SHA] and cryptographic key sizes [384-bit] that meets the following: [ISO/IEC 18031(2011)].
.

6.1.1.3 User Data Protection

FDP_IFC.1 Subset Information Flow Control

Hierarchical to: No other components.
Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the [Information Flow Blocking Policy] on [

- Subject list: User PC
- Information list: destination IP address and URL
- Operation list: detection, isolation, permission].

* Application notes: The operation (default value: isolation) details are as follows:

- Detection: Without information flow control, an alarm (SMS and email) is sent to the authorized administrator after audit record generation.
- Block: The connection is blocked by sending a RESET packet to the user PC, and an alarm (SMS and email) is sent to the authorized administrator after audit record generation.
- Permission: The information flow is not controlled, related audit record is not generated, and an alarm is not sent.

FDP_IFF.1 Simple security attributes

Hierarchical to: No other components.
Dependencies: FDP_IFC.1 Subset Information Flow Control
FMT_MSA.3 Static attribute initialization

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

- Subject: user PC
- Subject security attribute: IP address
- Information: Traffic Transmission from user PC to external network
- Information security attribute: destination IP and destination URL].

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: [

- When the destination IP/URL of the network packet Transmission from user PC is compared with the unauthorized server list (IP/URL), and the destination IP/URL is not found in the unauthorized server list].

FDP_IFFF.1.3 The TSF shall enforce the [none].
FDP_IFF.1.4 The TSF shall explicitly authorize an information flow based on the following rules: [none].
FDP_IFF.1.5 The TSF shall explicitly deny an information flow based on the following rules: [none].

6.1.1.4 Identification and Authentication

FIA_AFL.1 Authentication Failure Handling

Hierarchical to: No other components.
Dependencies: FIA_UAU.1 Timing of Authentication

FIA_AFL.1.1 The TSF shall detect when *[5]* unsuccessful authentication attempts occur related to [login attempts to the TOE security management web page].
FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been *met*, *surpassed*, the TSF shall [account lock for 10 min for top administrator and account lock for other administrators].

FIA_ATD.1 User Attribute Definition

Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to the **authorized administrator**: [ID, password, email address, right, alarm]

FIA_SOS.1 Verification of Secrets

Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [at least nine up to 15 characters including at least English upper and lower cases, numeric characters, and special characters (~, !, @, #, $, %, ^, & , *, (,), _, +)].

FIA_UAU.2 User Authentication Before Any Action

Hierarchical to: FIA_UAU.1 Timing of Authentication
Dependencies: FIA_UID.1 Timing of Identification

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

FIA_UAU.4 Single-use Authentication Mechanisms

Hierarchical to: No other components.
Dependencies: No dependencies.

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [ID/password-based user authentication].

FIA_UAU.7 Protected Authentication Feedback

Hierarchical to: No other components.
Dependencies: FIA_UAU.1 Timing of Authentication

FIA_UAU.7.1 The TSF shall provide only [showing password conversion into “*” on monitor, succeed and failure message] to the **administrator** while the authentication is in progress.

FIA_UID.2 User Identification Before Any Action

Hierarchical to: FIA_UID.1 Timing of Identification
Dependencies: No dependencies.

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

6.1.1.5 Security Management

FMT_MOF.1 Security Function Management

Hierarchical to: No other components.
Dependencies: FMT_SMR.1 Security Roles
FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1 The TSF shall restrict the ability to *disable, enable* the functions [each functional list in [Table 7]].

[Table 7] List of the Functions

Administrator role / Function list	Top administrator	Operation administrator
Identification and Authentication	disable, enable	disable, enable
Audit record generation	enable	enable
Audit review	disable, enable	disable, enable
Prediction and prevention of audit data loss	disable, enable	disable, enable
Information flow control	disable, enable	disable, enable
Self-test	enable	enable
Information Collection	disable, enable	disable, enable
Analysis on Collected Information	disable, enable	disable, enable
Reaction	disable, enable	disable, enable
Pattern management	enable	enable

FMT_MSA.1 Management of Security Attributes

Hierarchical to: No other components.
Dependencies: [FDP_ACC.1 Subset Access Control, or
FDP_IFC.1 Subset Information Flow Control]
FMT_SMR.1 Security Role
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1 The TSF shall enforce the [Information Flow Blocking Policy] to restrict the ability to *[none]* the security attributes [destination IP and destination URL] to [top administrator and operation administrator].

FMT_MSA.3 Static attribute initialization

Hierarchical to: No other components.
Dependencies: FMT_MSA.1 Management of Security Attributes
FMT_SMR.1 Security Roles

FMT_MSA.3.1 The TSF shall enforce the [Information Flow Blocking Policy] to provide *restrictive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2 The TSF shall allow the [top administrator and operation administrator] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1 TSF Data Management

Hierarchical to: No other components.
Dependencies: FMT_SMR.1 Security Roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to *query, modify, delete, and [add]* the [TSF data in [Table 8]] to [top administrator, operation administrator, read-only administrator, and monitoring administrator].

[Table 8] TSF Data

Administrator role / TSF data	Top Administrator	Operation Administrator	Read right administrator	Monitoring administrator
Security audit query	Query	Query	Query	-
Administrator connection setup	Addition, modification, deletion	Addition, modification, deletion	-	-
System setup	modification	modification	-	-
Security path setup	modification	modification	-	-
Live update setup	modification	modification	-	-
Initialization	modification	modification	-	-
ZombieZERO Agent policy setup	Query, addition, modification, deletion	Query, addition, modification, deletion	-	-

ZombieZERO Detector policy setup	modification	modification	-	-
Pattern management	Query, addition, modification, deletion	Query, addition, modification, deletion	-	-

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.
Dependencies: No dependencies.

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

- TSF function management specified in FMT_MOF.1
- Management of Security Attributes specified in FMT_MSA.1
- Static attribute management specified in FMT_MSA.3
- TSF data management specified in FMT_MTD.1

]

FMT_SMR.1 Security Roles

Hierarchical to: No other components.
Dependencies: FIA_UID.1 Timing of Identification

FMT_SMR.1.1 The TSF shall maintain the roles [top administrator, operation administrator, read-only administrator, and monitoring administrator].
FMT_SMR.1.2 The TSF shall be able to associate **authorized administrators** with roles.

* Application notes: The roles of the authorized administrators are as follows:

Classification	Role
Top administrator	▪ Top administrator is an authorized administrator with full privileges. Top administrator can use all the functions specified in security management, and can add / delete administrators and assign / change roles.
Operation administrator	▪ Operation administrator is an authorized administrator with full privileges. Operation administrator can use all the functions specified in security management, and can add / delete administrators except top privilege and assign / change roles.
Read-only administrator	▪ Read-only administrator can access a monitoring menu and search audit records.

Monitoring administrator	▪ Monitoring administrator can access the monitoring menu.

6.1.1.6 TSF Protection

FPT_TST.1(1) TSF testing

Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_TST.1.1 The TSF shall run a suite of self tests *during initial start-up, periodically during normal operation, and at the request of the authorized administrator* to demonstrate the correct operation of *the TSF*.
FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of *[none]*.
FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of [*TSF execution code*].

6.1.1.7 TOE Access

FTA_SSL.3 TSF-initiated Termination

Hierarchical to: No other components.
Dependencies: No dependencies.

FTA_SSL.3.1 The TSF shall terminate an interactive session after a [timeout value set by authorized administrator (default: 10 min) or designated interval (5-10 min)].

6.1.1.8 Information Collection

DSM_COL.1 file collection

Hierarchical to: No other components
Dependencies: No dependencies.

DSM_COL.1.1 TSF shall extract and store the files whose file extension is one of [EXE, DOC, DOCX, XLS, XLSX, PPT, PPTX, HWP, PDF, TAR, 7Z, ZIP, XZ, GZ, RAR, DLL, SWF, XLSM, XLSB, HTM HTML,

XLTX, XLTM, XLT, CSV, PRN, DIF, SLK, XLAM, XLA, XPS, ODS, XLW, DOCM, DOTX, DOT, RTF, ODT, WPS, PPTM, POTX, POT, THMX, PPSX, PPSM, PPS, PPAM, PPA, MP4, WMV, EMF, ODP, PNG, JPG, JPEG, and GIF] over the following protocols [HTTP, POP3, SMTP, IMAP, and FTP] to detect malicious code.
DSM_COL.1.2 The TSF shall generate information of [collection time, network protocol, file name, size, source IP/port, and destination IP/port] for the collected files.

6.1.1.9 Information Analysis

DSM_INA.1 Information Analysis

Hierarchical to: No other components.
Dependencies: No dependencies.

DSM_INA.1.1 The TSF shall perform the following analysis [
a) whether collected files contain malicious code by comparing them with bitdefender pattern;
b) whether collected files contain malicious code by comparing them with Yara rules; and
c) whether collected files contain malicious code by comparing them with action information of collected files in the Analyzer Agent and Yara rules,] based on collected data.
DSM_INA.1.2 TSF shall generate the following information after analyzing the collected data:
a) Analysis end date, analysis-required file name and extension, file size, and analysis result (malicious or not); and
b) [none].

6.1.1.10 Reaction

DSM_RCT.1(1) Security Reaction

Hierarchical to: No other components.
Dependencies: DSM_INA.1 information analysis
DSM_RCT.1.1 The TSF shall perform the [registration of the discovered malicious code list and sending the reaction policy (detection/block/isolation) to ZombieZERO Agent] when malicious behaviors and code are *detected*.

6.1.2 Analyzer Agent

6.1.2.1 Information Collection

DSM_COL.2 file execution information collection

Hierarchical to: No other components.
Dependencies: No dependencies.

DSM_COL.2.1 The TSF shall execute files with the following extensions: [EXE, DOC, XLS, PPT, HWP, PDF, DLL, SWF, XLSM, XLTM, XLAM, DOCM, PPTM, JPG, JPEG, GIF, and JS] to detect malicious behaviors.
DSM_COL.2.2 After the TSF monitors the executed files, it shall record the following behavior records [
a) registry-related action: creation, modification, or deletion of registry key (path), registry name, and registry value (data);
b) network-related action: network communication (connection to IP/Port);
c) memory-related action: memory write and read;
d) file-related action: new file creation, existing file modification, and existing file deletion; and
e) process-related action: process creation and termination
] as a file form.

6.1.3 ZombieZERO Agent

6.1.3.1 Reaction

DSM_RCT.1(2) Security Reaction

Hierarchical to: No other components.
Dependencies: DSM_INA.1 information analysis

DSM_RCT.1.1 The TSF shall perform the detection, block, or isolation according to the reaction policy (detection, block, or isolation) transferred by ZombieZERO Detector and sending the exe extension file that are not found in the reaction policy to ZombieZERO Detector] when malicious behavior and code *reaction policy is received*.

* Application notes: The reaction functions of ZombieZERO Agent are as follows:

- ✓ Detection: A process is detected in the black list from the processes executed in the internal user PC, and the related information is then sent to ZombieZERO Detector without block and isolation.
- ✓ Block: A process executed in the user PC is terminated.
- ✓ Isolation: After a process executed in the user PC is terminated, a file is moved to a designated quarantine folder.

6.1.3.2 Cryptographic Support

FCS_CKM.4(2) Cryptographic key destruction

Hierarchical to: No other components.
Dependences: [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 [overwrite with zeroization '0'] that meets the following: [none].

FCS_COP.1(4) Cryptographic operation

Hierarchical to: No other components.
Dependences: [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 [Operating decryption of malicious code blocking/isolate policy files] in accordance with a specified cryptographic algorithm [AES CBC Mode] and cryptographic key sizes [128-bit] that meets the following: [Section 5.2 in ISO/IEC 18033-3:2010].

FCS_COP.1(5) Cryptographic operation

Hierarchical to: No other components.
Dependences: [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 [ZombieZERO Agent registry value hash generations] in accordance with a specified cryptographic algorithm [SHA] and cryptographic key sizes [384-bit] that meets the following: [ISO/IEC 18031(2011)].

6.1.3.3 TSF Protection

FPT_TST.1(2) TSF testing

Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_TST.1.1 The TSF shall run a suite of self-tests *during initial start-up and periodically during normal operation* to demonstrate the correct operation of the *TSF*.
FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of [*the setup value*].
FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of [*none*].

6.2 Security Assurance Requirements for TOE

The Evaluation Assurance Levels (EALs) is EAL2. Security assurance requirements are taken from the Common Criteria Part 3. The assurance requirements are listed in Table 9.

[Table 9] Security Assurance Requirements

Class	Assurance requirements	
ADV: Development	ADV_ARC.1	Security Architecture Description
	ADV_FSP.2	Security-enforcing functional specification
	ADV_TDS.1	Basic design
AGD: Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
ALC: Life-Cycle Support	ALC_CMC.2	Use of a CM system
	ALC_CMS.2	Parts of the TOE CM Coverage
	ALC_DEL.1	Delivery procedures
ASE: Security Target Evaluation	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Component 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

6.2.1 Class ASE: Security Target Evaluation

6.2.1.1 ASE_INT.1 ST introduction

Dependencies: No dependencies.

Developer action elements:
ASE_INT.1.1D The developer shall provide an ST introduction.

Content and presentation elements:
ASE_INT.1.1C The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
ASE_INT.1.2C The ST reference shall uniquely identify the ST.
ASE_INT.1.3C The TOE reference shall uniquely identify the TOE.
ASE_INT.1.4C The TOE overview shall summarize the usage and major security features of the TOE.
ASE_INT.1.5C The TOE overview shall identify the TOE type.
ASE_INT.1.6C The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.
ASE_INT.1.7C The TOE description shall describe the physical scope of the TOE.
ASE_INT.1.8C The TOE description shall describe the logical scope of the TOE.

Evaluator action elements:
ASE_INT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_INT.1.2E The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

6.2.1.2 ASE_CCL.1 Conformance Claims

Dependencies:
ASE_INT.1 ST introduction
ASE_ECD.1 Extended Component Definition
ASE_REQ.1 Stated security requirements

Developer action elements:
ASE_CCL.1.1D The developer shall provide conformance claims.
ASE_CCL.1.2D The developer shall provide a conformance claim rationale.

Content and presentation elements:
ASE_CCL.1.1C The conformance claims shall contain a CC conformance claim that identifies the version of the CC to which the ST and TOE claim conformance.
ASE_CCL.1.2C The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.
ASE_CCL.1.3C The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.
ASE_CCL.1.4C The CC conformance claim shall be consistent with the extended component definition.
ASE_CCL.1.5C The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.
ASE_CCL.1.7C The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.
ASE_CCL.1.8C The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.
ASE_CCL.1.9C The conformance claim rationale shall demonstrate that the statement of the security objectives is consistent with the statement of the security objectives in the PPs for which conformance is being claimed.
ASE_CCL.1.10C The conformance claim rationale shall demonstrate that the statement of the security requirements is consistent with the statement of the security requirements in the PPs for which conformance is being claimed.

Evaluator action elements
ASE_CCL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.1.3 ASE_SPD.1 Security Problem Definition

Dependencies: No dependencies.

Developer action elements
ASE_SPD.1.1D The developer shall provide the security problem definition.

Content and presentation elements:
ASE_SPD.1.1C The security problem definition shall describe the threats.
ASE_SPD.1.2C All threats shall be described in terms of a threat agent, an asset, and an adverse action.
ASE_SPD.1.3C The security problem definition shall describe the OSPs.
ASE_SPD.1.4C The security problem definition shall describe the assumptions about TOE operational environment of the TOE.

Evaluator action elements
ASE_SPD.1.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.2.1.4 ASE_OBJ.2 Security Objectives

Dependencies: ASE_SPD.1 Security Problem Definition

Developer action elements
ASE_OBJ.2.1D The developer shall provide a statement of security objectives.

ASE_OBJ.2.2D The developer shall provide a security objective rationale.

Content and presentation elements:
ASE_OBJ.2.1C The statement of security objectives shall describe the security objectives for the TOE and the security objectives for the operational environment.
ASE_OBJ.2.2C The security objectives rationale shall trace each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective.
ASE_OBJ.2.3C The security objectives rationale shall trace each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective.
ASE_OBJ.2.4C The security objectives rationale shall demonstrate that the security objectives counter all threats.
ASE_OBJ.2.5C The security objectives rationale shall demonstrate that the security objectives enforce all OSPs.
ASE_OBJ.2.6C The security objectives rationale shall demonstrate that the security objectives for the operational environment uphold all assumptions.

Evaluator action elements:
ASE_OBJ.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

6.2.1.5 ASE_ECD.1 Extended Component Definition

Dependencies: No dependencies.

Developer action elements
ASE_OBJ.2.1D The developer shall provide a statement of security requirements.
ASE_ECD.1.2D The developer shall provide an extended components definition.

Content and presentation elements:
ASE_ECD.1.1C The statement of security requirements shall identify all extended security requirements.
ASE_ECD.1.2C The extended components definition shall define an extended component for each extended security requirement.
ASE_ECD.1.3C The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.
ASE_ECD.1.4C The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.
ASE_ECD.1.5C The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

Evaluator action elements

ASE_ECD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ASE_ECD.1.2E The evaluator shall confirm that no extended component can be clearly expressed using existing components.

6.2.1.6 ASE_REQ.2 Derived Security Requirements

Dependencies:
ASE_OBJ.2 Security Objectives
ASE_ECD.1 Extended Component Definition

Developer action elements
ASE_REQ.2.1D The developer shall provide a statement of security requirements.
ASE_REQ.2.2D The developer shall provide a security requirements rationale.

Content and presentation elements:
ASE_REQ.2.1C The statement of security requirements shall describe the SFRs and the SARs.
ASE_REQ.2.2C All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.
ASE_REQ.2.3C The statement of security requirements shall identify all operations on the security requirements.
ASE_REQ.2.4C All operations shall be performed correctly.
ASE_REQ.2.5C Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.
ASE_REQ.2.6C The security requirements rationale shall trace each SFR back to the security objectives for the TOE.
ASE_REQ.2.7C The security requirements rationale shall demonstrate that the SFRs meet all security objectives for the TOE.
ASE_REQ.2.8C The security requirements rationale shall explain why the SARs were chosen.
ASE_REQ.2.9C The statement of security requirements shall be internally consistent.

Evaluator action elements
ASE_REQ.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.1.7 ASE_ TSS.1 TOE Summary Specification

Dependencies:
ASE_INT.1 ST introduction
ASE_REQ.1 Stated security requirements
ADV_FSP.1 Basic functional specification

Developer action elements
ASE_TSS.1.1D The developer shall provide a TOE summary specification.

Content and presentation elements:
ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR.

Evaluator action elements
ASE_TSS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ASE_INT.1.2E The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

6.2.2 Class ADV : Development

6.2.2.1 ADV_ ARC.1 Security Architecture Description

Dependencies:
ADV_FSP.1 Basic functional specification
ADV_TDS.1 Basic design

Developer action elements
ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.
ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.
ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

Content and presentation elements:
ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.
ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.
ADV_ARC.1.3C The security architecture description shall describe how the TSF initialisation process is secure.
ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.
ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.
Evaluator action elements

ADV_ARC.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.2.2 ADV_ FSP.2 Security-enforcing functional specification

Dependencies: ADV_TDS.1 Basic design

Developer action elements
ADV_FSP.2.1D The developer shall provide a functional specification.
ADV_FSP.2.2D The developer shall provide a tracing from the functional specification to the SFRs.
Content and presentation elements:
ADV_FSP.2.1C The functional specification shall completely represent the TSF.
ADV_FSP.2.2C The functional specification shall describe the purpose and method of use for all TSFI.
ADV_FSP.2.3C The functional specification shall identify and describe all parameters associated with each TSFI.
ADV_FSP.2.4C For each SFR-enforcing TSFI, the functional specification shall describe the SFR-enforcing actions associated with the TSFI.
ADV_FSP.2.5C For each SFR-enforcing TSFI, the functional specification shall describe direct error messages resulting from processing associated with the SFR-enforcing actions.
ADV_FSP.2.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements
ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

6.2.2.3 ADV_ TDS.1 Basic design

Dependencies: ADV_FSP.2 Security-enforcing functional specification

Developer action elements
ADV_TDS.1.1D The developer shall provide the design of the TOE.
ADV_TDS.1.2D The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

Content and presentation elements:
ADV_TDS.1.1C The design shall describe the structure of the TOE in terms of subsystems.
ADV_TDS.1.2C The design shall identify all subsystems of the TSF.
ADV_TDS.1.3C The design shall provide the behaviour summary of each SFR-supporting or SFR-non-interfering TSF subsystem.
ADV_TDS.1.4C The design shall summarise the SFR-enforcing behaviour of the SFR-enforcing subsystems.

ADV_TDS.1.5C The design shall provide a description of the interactions among SFR-enforcing subsystems of the TSF, and between the SFR-enforcing subsystems of the TSF and other subsystems of the TSF.
ADV_TDS.1.6C The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.

Evaluator action elements
ADV_TDS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_TDS.1.2E The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

6.2.3 Class AGD : Guidance Documents

6.2.3.1 ADV_ OPE.1 Operational user guidance

Dependencies: ADV_FSP.1 Basic functional specification

Developer action elements
AGD_OPE.1.1D The developer shall provide operational user guidance.
Content and presentation elements:
AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.
AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.
AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.
AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.
AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements
AGD_OPE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.3.2 ADV_ PRE.1 Preparative procedures

Dependencies: No dependencies.

Developer action elements
AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements:
AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements
AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

6.2.4 Class ALC : Life-Cycle Support

6.2.4.1 ALC_ CMC.2 Use of a CM system

Dependencies: ALC_CMS.1 TOE CM Coverage

Developer action elements
ALC_CMC.2.1D The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.2.2D The developer shall provide the CM documentation.
ALC_CMC.2.3D The developer shall use a CM system.

Content and presentation elements:
ALC_CMC.2.1C The TOE shall be labelled with its unique reference.
ALC_CMC.2.2C The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.2.3C The CM system shall uniquely identify all configuration items.

Evaluator action elements

ALC_CMC.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.4.2 ALC_ CMS.2 Parts of the TOE CM coverage

Dependencies: No dependencies.

Developer action elements
ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:
ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; and the parts that comprise the TOE.
ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.
ALC_CMS.2.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

Evaluator action elements
ALC_CMS.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.4.3 ALC_ DEL.1 Delivery procedures

Dependencies: No dependencies.

Developer action elements
ALC_DEL.1.1D The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.
ALC_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation elements:
ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

Evaluator action elements
ALC_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.5 Class ATE : Tests

6.2.5.1 ATE_ COV.1 Evidence of coverage

Dependencies:
ADV_FSP.2 Security-enforcing functional specification
ATE_FUN.1 Functional testing

Developer action elements
ATE_COV.1.1D The developer shall provide evidence of the test coverage.

Content and presentation elements:
ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

Evaluator action elements
ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.5.2 ATE_ FUN.1 Functional testing

Dependencies: ATE_COV.1 Evidence of test coverage

Developer action elements
ATE_FUN.1.1D The developer shall test the TSF and document the results.
ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation elements:
ATE_FUN.1.1C The test documentation shall consist of test plans, expected test results and actual test results.
ATE_FUN.1.2C The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.3C The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.4C The actual test results shall be consistent with the expected test results.

Evaluator action elements
ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

6.2.5.3 ATE_ IND.2 Independent Testing - Sample

Dependencies:
ADV_FSP.2 Security-enforcing functional specification
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures
AGD_COV.1 Evidence of test coverage
ATE_FUN.1 Functional testing

Developer action elements
ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation elements:
ATE_IND.2.1C The TOE shall be suitable for testing.
ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements
ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.
ATE_IND.2.3E The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

6.2.6 Class AVA : Vulnerability assessment

6.2.6.1 AVA_ VAN.2 Vulnerability Analysis

Dependencies:
ADV_ARC.1 Security Architecture Description
ADV_FSP.2 Security-enforcing functional specification
ADV_TDS.1 Basic design
AGD_OPE.1 Operational user guidance
AGD_PRE.1 Preparative procedures

Developer action elements
AVA_VAN.2.1D The developer shall provide the TOE for testing.

Content and presentation elements:
AVA_VAN.2.1C The TOE shall be suitable for testing.

Evaluator action elements

AVA_VAN.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.2.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.2.3E The evaluator shall perform an independent vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design and security architecture description to identify potential vulnerabilities in the TOE.

AVA_VAN.2.4E The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6.3 Security Requirements Rationale

Security requirements rationale demonstrates that described IT security functional required components are suitable for Security objectives, and they are appropriate for dealing with security risks.

6.3.1 Security Functional Requirements Rationale

The security functional requirements rationale proofs the list down below.
Each of TOE security objectives is treated by at least one TOE security functional requirement.
Each of TOE security functional requirements deals with at least one TOE security objective.

[Table 10] Mapping of Security Objectives and Security Requirements

	O. Audit	O. Audit management	O. Identification and authentication	O.Secure Cryptographic	O.Stored Data Protection	O. Detection and reaction to malicious code	O. Blocking unauthorized information flow	O. Self-protection
FAU_GEN.1	✓							
FAU_GEN.2	✓							
FAU_SAR.1	✓							
FAU_SAR.3	✓							
FAU_STG.1	✓							
FAU_STG.3	✓							
FAU_STG.4	✓							
FCS_CKM.1				✓				
FCS_CKM.4(1)		✓		✓	✓			
FCS_CKM.4(2)				✓	✓			
FCS_COP.1(1)				✓	✓			

	O. Audit	O. Audit management	O. Identification and authentication	O.Secure Cryptographic	O.Stored Data Protection	O. Detection and reaction to malicious code	O. Blocking unauthorized information flow	O. Self-protection
FCS_COP.1(2)		✓		✓				
FCS_COP.1(3)			✓	✓				✓
FCS_COP.1(4)				✓	✓			
FCS_COP.1(5)				✓				✓
FDP_IFC.1							✓	
FDP_IFF.1							✓	
FIA_AFL.1			✓					
FIA_ATD.1			✓					
FIA_SOS.1			✓					
FIA_UAU.2			✓					
FIA_UAU.4			✓					
FIA_UAU.7			✓					
FIA_UID.2			✓					
FMT_MOF.1		✓						
FMT_MSA.1		✓						
FMT_MSA.3		✓						
FMT_MTD.1		✓						
FMT_SMF.1		✓						
FMT_SMR.1		✓						
FPT_TST.1(1)								✓
FPT_TST.1(2)								✓

	O. Audit	O. Audit management	O. Identification and authentication	O.Secure Cryptographic	O.Stored Data Protection	O. Detection and reaction to malicious code	O. Blocking unauthorized information flow	O. Self-protection
FTA_SSL.3		✓						
DSM_COL.1						✓		
DSM_COL.2						✓		
DSM_INA.1						✓		
DSM_RCT.1(1)						✓		
DSM_RCT.1(2)						✓		

FAU_GEN.1 Audit Data Generation

This component meets TOE security objective of O. Audit because it defines audit target events and assures audit record generation ability.

FAU_GEN.2 User Identity Association

This component meets TOE security objective of O. Audit because it is required to guarantee the ability to associate a user who incurs an event and audit target.

FAU_SAR.1 Audit Review

This component meets TOE security objective of O. Audit because it assures authorized administrator's ability to examine audit record.

FAU_SAR.3 Selectable Audit Review

This component meets TOE security objective of O. Audit because it assures audit data searching and sorting ability by logical connection standard.

FAU_STG.1 Protected Audit Trail Storage

This component meets TOE security objective of O. Audit because it assures audit record protection ability against inappropriate modify and delete.

FAU_STG.3 Action in case of Possible Audit Data Loss

This component meets TOE security objective of O. Audit because it assures counter action ability when audit trail exceeds the pre-defined threshold.

FAU_STG.4 Prevention of Audit Data Loss

This component meets TOE security objective of O. Audit because it assures counter action ability when audit storage is full.

FCS_CKM.1 Cryptographic key generation

This component meets the Security Objective “O. Secure Password” because ZombieZERO Detector assures the encryption key (128-bit, CBC Mode) generation using the AES (128-bit) encryption algorithm.

FAU_CKM.4(1) Cryptographic key destruction

This component meets the Security Objective “O.Security Management”, “O.Secure Password”, and “O.Stored Data Protection” since it assures a method to destroy an encryption keys used in cryptographic key operation of security policy files and signature verification of update files in ZombieZERO Detector by overwriting them with random numbers.

FAU_CKM.4(2) Cryptographic key destruction

This component meets the Security Objective “O.Secure Password”, and O.Stored Data Protection” because it assures a method to destroy an encryption key by overwriting the encryption keys used in cryptographic operation of setup values of ZombieZERO Agent with “0”.

FAU_COP.1(1) Cryptographic operation

This component meets the Security Objective “O.Secure Password”, and O.Stored Data Protection” because it assures cryptographic operations on malicious code blocking/isolate policy files with the secure encryption algorithm (AES128, CBC Mode) that complies with the security objectives of ZombieZERO Detector.

FAU_COP.1(2) Cryptographic operation

This component meets the Security Objective “O.Security Management” and “O.Secure Password” because it assures signature verifications on update files by using the secure encryption algorithm (RSA2048) that complies with the security objectives of ZombieZERO Detector.

FAU_COP.1(3) Cryptographic operation

This component meets the Security Objective “O.Identification and Authentication, “O.Secure Password”, and “O.Self-protection” because it assures the execution code and administrator password has generation of ZombieZERO Detector by using the secure encryption algorithm (SHA384) that complies with the security objectives of ZombieZERO Detector.

FAU_COP.1(4) Cryptographic operation

This component meets the Security Objective “O.Secure Password”, and O.Stored Data Protection” because it assures encryption and decryption operations on encryption keys and cryptographic operations on malicious code blocking/isolate policy files by using the secure encryption algorithm (AES128, CBC Mode) that complies with the security objectives of ZombieZERO Agent.

FAU_COP.1(5) Cryptographic operation

This component meets the Security Objective “O.Secure Password” and “O.Self-protection” because it assures the hash generation of registry values of ZombieZERO Agent by using the secure encryption algorithm (SHA384) that complies with the security objectives of ZombieZERO Agent.

FDP_IFC.1 Subset Information Flow Control

This component meets TOE security objective of O. Access to Unauthorized External Network because it assures the information flow control from user PCs to Unauthorized IP/URL list (ex. malicious bots, malicious bot distribution servers, and C&C servers)

FDP_IFF.1 Simple security attributes

This component meets TOE security objective of O. Access to Unauthorized External Network because it assures the information flow control from user PCs to Unauthorized IP/URL list(ex. malicious bots, malicious bot distribution servers, and C&C servers) according to the IP/URL block policy set by the authorized administrator.

FIA_AFL.1 Authentication Failure Handling

This component meets TOE security objective of O. Identification and Authentication because it assures providing a function that locks the account of Top Administrator for 10 min and locks other administrators accounts if the number of user authentication failures exceeds five times.

FIA_ATD.1 User Attribute Definition

This component meets TOE security objective of O. Identification and Authentication because it defines the security attribute list (ID, password, email address, right, alarm) by authorized administrator.

FIA_SOS.1 Password Verification

This component meets TOE security objective of O. Identification and Authentication because it provides a mechanism to verify whether a password has at least nine characters up to 15 characters including at least one alphabet upper case, one lower case, one number and one special character).

FIA_UAU.2 User Authentication Before Any Action

This component meets TOE security objective of O. Identification and Authentication because it assures the successful authentication of users who like to access the TOE security management.

FIA_UAU.4 Single-use Authentication Mechanisms

This component meets TOE security objective of O. Identification and Authentication because it assures the single-use of authentication data in relation to administrator's authentication information.

FIA_UAU.7 Authentication Feedback Protection

This component meets TOE security objective of O. Identification and Authentication because it assures providing only authentication feedback designated to users while the authentication is underway.

FIA_UAU.2 User Identification Before Any Action

This component meets TOE security objective of O. Identification and Authentication because it assures the successful identification of users who like to access the TOE security management.

FMT_MOF.1 Management of Security Functions Behavior

This component meets TOE security objective of O. Security Management because it assures the ability the manage security functions by authorized administrators.

FMT_MSA.1 Management of Security Attributes

This component meets TOE security objective of O. Security Management because it provides the ability to change security attributes by authorized administrators.

FMT_MSA.3 Static attribute initialization

This component meets TOE security objective of O. Security Management because it assures providing a limited default of security attribute in the IP/URL block policy and initialization function by authorized administrators.

FMT_MTD.1 Management of TSF Data

This component meets TOE security objective of O. Security Management because it provides the ability to maintain and manage TSF data up to the latest state by authorized administrators.

FMT_SMF.1 Specification of Management Functions

This component meets TOE security objective of O. Security Management because it assures the ability the manage security functions by authorized administrators.

FMT_SMR.1 Security Role

This component meets TOE security objective of O. Security Management because it assures user association into authorized administrator role.

FPT_TST.1 TSF(1) Testing

This component meets TOE Security objective of O. TSF Data Protection because it assures providing a data integrity verification function for TSF data at the time of startup, upon the request from the authorized administrator, and during operations periodically.

FPT_TST.1 TSF(2) Testing

This component meets TOE Security objective of O. TSF Data Protection because it assures data integrity verification function for setup values at the time of startup and during operations periodically.

FTA_SSL.3 TSF-initiated Termination

This component meets TOE Security objective of O. TSF Data Protection because it terminates a session of administrator after a certain inactive time (five to 10 min, default value is 10 min).

DSM_COL.1 file collection

This component meets TOE Security objective of O. Malicious Code Detection and Reaction because ZombieZERO Detector assures information collection through extraction and storage of 'EXE, DOC, DOCX, XLS, XLSX, PPT, PPTX, HWP, PDF, TAR, 7Z, ZIP, XZ, GZ, RAR, DLL, SWF, XLSM, XLSB, HTM HTML, XLTX, XLTM, XLT, CSV, PRN, DIF, SLK, XLAM, XLA, XPS, ODS, XLW, DOCM, DOTX, DOT, RTF, ODT, WPS, PPTM, POTX, POT, THMX, PPSX, PPSM, PPS, PPAM, PPA, MP4, WMV, EMF, ODP, PNG, JPG, JPEG, GIF' over the protocols [HTTP, POP3. SMTP, IMAP, FTP].

DSM_COL.2 file execution information collection

This component meets TOE Security objective of O. Malicious Code Detection and Reaction because Analyzer Agent assures information collection of files by executing collected files.

DSM_INA.1 Information Analysis

This component meets TOE Security objective of O. Malicious Code Detection and Reaction because it performs information analysis based on collected data, and assures storing record and event information of detected security violation from data generated during the analysis.

DSM_RCT.1(1) Security Reaction

This component meets TOE Security objective of O. Malicious Code Detection and Reaction because ZombieZERO Detector ensures the transmission of reaction policy (detection, block, or isolation) to ZombieZERO Agent at the time of malicious behavior or code detection based on the analysis results on collected data.

DSM_RCT.1(2) Security Reaction

This component meets TOE Security objective of O. Malicious Code Detection and Reaction because ZombieZERO Agent ensures performing a reaction against suspicious malicious files based on the reaction policy Transmission from ZombieZERO Detector and storing security violation event.

6.3.2 Security Assurance Requirements Rationale

This ST satisfies the EAL 2 by accommodating the assurance package of EAL2.
The TOE can provide the assurance that it has functional and interface specifications, descriptions, basic description of TOE structure, testing, and resistance to basic attack success possibilities by vulnerability analysis (based on provided functional specifications, TOE design, security architecture description, and description evidence). It can also provide the assurance through the CM system and evidence of delivery procedures in a secured manner.

The assurance means that satisfies the requirements of the EAL2 package is described in the Assurance Document, which is referred to in Section 6.2. Each document is sufficient to satisfy the assurance requirement.

[Table 11] Assurance Requirements - EAL2

Class	Assurance requirements
Development	ADV_ARC.1 Security Architecture Description
	ADV_FSP.2 Security-enforcing functional specification
	ADV_TDS.1 Basic design
Guidance Documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
Life-cycle support	ALC_CMC.2 Use of CM system
	ALC_CMS.2 TOE CM coverage
	ALC_DEL.1 Delivery procedures
Security Target Evaluation	ASE.CCL.1 Conformance Claims
	ASE_ECD.1 Extended Component 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
Tests	ATE_COV.1 Evidence of test coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent test : Sample test
Vulnerability Evaluation	AVA_VAN.2 Vulnerability Analysis

6.3.3 Rationale Dependencies

6.3.3.1 Rationale for SFR's Dependencies

[Table 12] SFR's Dependencies

No.	SFRs	Dependencies	Reference No.
1	FAU_GEN.1	FPT_STM.1	-[1)]
2	FAU_GEN.2	FAU_GEN.1 FAI_UID.1	1 (24)[2)]
3	FAU_SAR.1	FAU_GEN.1	1
4	FAU_SAR.3	FAU_SAR.1	3
5	FAU_STG.1	FAU_GEN.1	2
6	FAU_STG.3	FAU_STG.1	5
7	FAU_STG.4	FAU_STG.1	5
8	FCS_CKM.1	[FCS_CKM.2 or FCS.COP.1] FCS_CKM.4	11, 14 9, 10
9	FCS_CKM.4(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	8
10	FCS_CKM.4(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	8
11	FCS_COP.1(1)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	8 9
12	FCS_COP.1(2)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	-[4)] 9
13	FCS_COP.1(3)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	-[5)] 9
14	FCS_COP.1(4)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	8 10
15	FCS_COP.1(5)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	-[5)] 10
16	FDP_IFC.1	FDP_IFF.1	17
17	FDP_IFF.1	FDP_IFC.1	16

No.	SFRs	Dependencies	Reference No.
		FMT_MSA.3	27
18	FIA_AFL.1	FIA_UAU.1	(21)[3)]
19	FIA_ATD.1	-	-
20	FIA_SOS.1	-	-
21	FIA_UAU.2	FIA_UID.1	(24)[2)]
22	FIA_UAU.4	-	-
23	FIA_UAU.7	FIA_UAU.1	(21)[3)]
24	FIA_UID.2	-	-
25	FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	29 30
26	FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	17 29 30
27	FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	26 30
28	FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	29 30
29	FMT_SMF.1	-	-
30	FMT_SMR.1	FIA_UID.1	(24)[2)]
31	FPT_TST.1(1)	-	-
32	FPT_TST.1(2)		-
33	FTA_SSL.3	-	-
34	DSM_COL.1	-	-
35	DSM_COL.2	-	-
36	DSM_INA.1	-	-
37	DSM_RCT.1(1)	DSM_INA.1	36
38	DSM_RCT.1(2)	DSM_INA.1	36

1) FPT_STM.1, which is a dependence of Security Functional Requirement of FAU_GEN.1, is satisfied by Security Objective OE.Timestamp with regard to operational environment because

operational environment provides the reliable timestamp rather than TOE.

2) FAU_GEN.2, FIA_UAU.2, and FMT_SMR.1 have a dependence on FIA_UID.1, but they are satisfied with FIA_UID.2, which is hierarchical to FIA_UID.1.
3) FIA_AFL.1 and FIA_UAU.7 have a dependence on FIA_UAU.1, but they are satisfied with FIA_UAU.2, which is hierarchical to FIA_UAU.1.
4) [FDP_ITC.1, FDP_ITC.2, or FCS_CKM.1], which is a dependence of security function requirements of FCS_COP.1(2), is satisfied by the security objective “OE.Secure Update” of the operational environment because the TOE does not generate a public key used to verify electronic signatures, but a public key is provided by the operational environment.
5) FCS_COP.1(3) and FCS_COP.1(5) are satisfied because the used hash algorithm (SHA 384) does not use encryption keys although they have a dependence on [FDP_ITC.1, FDP_ITC.2, or FCS_CKM.1] and FCS_CKM.4.

6.3.3.2 SAR’s Dependencies

The dependency of each assurance package (EAL2) provided by the Common Criteria for Information Protection Systems is already satisfied.

7. TOE Summary Specification

7.1 Security Audit

ZombieZERO Detector employs reliable timestamps provided by the operational environment of ZombieZERO Detector at the event occurrence time to ensure that audit data are generated sequentially when audit data are generated. The operational environment (OS) of ZombieZERO Detector provides timestamps by periodically comparing the value stored in the real-time clock (RTC), which is hardware operated by ZombieZERO Detector.
ZombieZERO Detector stores all logs generated in the allocated storage (My-SQL) for audit date during the TOE operation, and provides search, statistics, and management functions.

7.1.1 Audit Data Generation

ZombieZERO Detector generates audit data generated during operation. Audit data generate and store audit records about matters such as information collection and information analysis to detect malicious behaviors and manage TOE security. (FAU_GEN.1)(FAU_GEN.2)

7.1.2 Security Audit Inquiry

ZombieZERO Detector provides a function to differentiate and review all TOE audit data according to the event type for top administrator, operation administrator, and read-only administrator. The TOE manages audit data through the file system (DBMS) of the operational environment of ZombieZERO Detector. ZombieZERO Detector reads files from the operational environment of ZombieZERO Detector upon a request from authorized administrator and provides an outcome for authorized administrator to analyze a result easily. In addition, security audit inquiry is not provided to monitoring administrator. (FAU_SAR.1), (FAU_SAR.3)

7.1.3 Audit Trail Protection

ZombieZERO Detector uses a DBMS, which is a repository in the operational environment of ZombieZERO Detector, as an audit trail storage. ZombieZERO Detector stores audit trails based on the file system provided by the operational environment of ZombieZERO Detector. ZombieZERO Detector

allows only authorized administrator to access audit trail records stored in the operational environment, and does not provide an interface to delete or modify audit records. Thus, TOE audit records are protected from unauthorized deletion or modification (FAU_STG.1).

7.1.4 Reaction and Prevention of Audit Data Loss

ZombieZERO Detector provides a function of reaction and prevention of audit data loss. Authorized administrators can set a limit of available space in the audit storage.
ZombieZERO Detector sends an alarm through SMS or email set by authorized administrator if the available space reaches designated limit in the audit trail storage.
The audit storage saturation can also be set by authorized administrator who can also set a storage period of audit storage.
ZombieZERO Detector performs an overwrite on the oldest audit data first when the available space of the audit trail storage set by authorized administrator is full or the audit storage period is expired to prevent a loss of audit records, and alarms are sent through SMS or email set by authorized administrators (FAU_STG.3) (FAU_STG.4).

7.2 Cryptographic Support

ZombieZERO Detector generates an encryption key in accordance with encryption algorithms (AES 128, CBC Mode) for malicious code blocking/isolate policy file encryption and performs malicious code blocking/isolate policy file encryption in accordance with the generated encryption algorithm (AES 128, CBC Mode). It also performs encryption key destruction according to an encryption key destruction method (overwriting with random numbers) to prevent exposure of encryption keys. (FCS_CKM.1), (FCS_CKM.4(1), (FCS_COP.1(1))
ZombieZERO Agent also performs decryption of malicious code blocking/isolate policy files encrypted in ZombieZERO Detector in accordance with the encryption algorithm (AES 128) to apply the policy, and encryption key destruction according to the encryption key destruction method (overwriting with “0”) to prevent the encryption key from being exposed.
(FCS_CKM.4(2), (FCS_COP.1(4))

ZombieZERO Detector generates hash code of execution code according to an encryption algorithm (SHA 384) at the time of first startup for self-testing. It also generates hash code of execution code according to an encryption algorithm (SHA 384) for self-testing in every test cycle (please refer to Section 7.6 TSF Protection).
(FCS_COP.1(3))
ZombieZERO Agent generates hash code of registry values according to an encryption algorithm (SHA 384) at the time of first startup for self-testing. It also generates hash code of registry values according to

an encryption algorithm (SHA 384) for self-testing in every test cycle (please refer to Section 7.6 TSF Protection).
(FCS_COP.1(5))

ZombieZERO Detector performs signature verifications on update files using a public key algorithm (RSA 2048) for signature verification of update file prior to applying an update file. It also performs public key destruction according to the encryption key destruction method (overwriting with random numbers) to prevent public keys from being exposed. (FCS_CKM.4(1), (FCS_COP.1(2))

Encryption algorithm specifications used in the TOE

ZombieZERO Detector

- Encryption of policy files: AES 128 CBC Mode(Section 5.2 in ISO/IEC 18033-3:2010)
- Update file signature verification: RSA 2048 (ISO/IEC 14888-2(2008))
- Hash generation of execution code and administrator password in ZombieZERO Detector: SHA 384 (ISO/IEC 18031(2011))

ZombieZERO Agent

- Decryption of policy files: AES 128 (Section 5.2 in ISO/IEC 18033-3:2010)
- Hash generation of registry values in ZombieZERO Agent: SHA 384 (ISO/IEC 18031(2011))

7.3 Blocking Information Flow

ZombieZERO Detector performs the “Information Flow Blocking SFP” by sending a RESET packet to an affected user PC according to that SFP if the destination is unauthorized IP/URL (e.g., malicious bot distribution server, C&C server) based on security attributes in the destination IP and URL once network packets in the user PCs over the internal network is leaked to the outside. If the security policy of ZombieZERO Detector is “detection”, audit records about security violation events that access unauthorized external networks (e.g., malicious bot distribution server and C&C server) are generated and all information flows are allowed without controlling in the user PC in an internal network. If the security policy is “permission”, all information flows of the user PC in the internal network are allowed without controlling and audit records are not generated. Thus, if detection or permission security policy is applied, “Information Flow Blocking Policy” is not enforced. Thus, authorized administrators shall take much care.
(FDP_IFC.1), (FDP_IFF.1)

7.4 Identification and Authentication

ZombieZERO Detector identifies all users who access the detector. No users who attempt an access can use any functions in the TOE until the user identification is complete. ZombieZERO Detector verifies ID, password, access right, and IP address information, which are registered to identify the authorized administrators, and then allows the access from the authorized administrators if the registered and input information are matched. The password information can be generated when the password creation criteria are met, and it shall be displayed mask characters to protect the authentication feedback when the password information is entered. It provides the authentication delay function according to the consecutive authentication failure handling function. The identification and authentication function is protected by using a timestamp to prevent re-using TOE authentication data.

7.5 Security Management

Top administrator can access the Security Management web page (GUI), which is the security management interface of the TOE, through a web browser (e.g., Internet Explorer 10/11) at the time of first startup once ZombieZERO Detector is normally installed. Here, top administrator can designate IP addresses that are allowed to connect to the Security Management web page (GUI). Afterward, ZombieZERO Detector allows connection to the security management interface (TLS1.2-based HTTPS) only when the administrator who attempts an access through one of the IPs that are allowed to access explicitly is successfully complete in the identification and authentication process enforced by ZombieZERO Detector.

7.5.1 Security Function Management

Roles of authorized administrators in the TOE are divided into four types: top administrator, operation administrator, read-only administrator, and monitoring administrator. The rights assigned to each of them are as follows: (FMT_MOF.1), (FMT_MSA.1), (FMT_MSA.3), (FMT_SMF.1) (FMT_SMR.1)

- Top administrator: An authorized administrator who has all the rights. He/she can employ all functions specified in the security management and can add and delete administrators as well as designate and change their roles. Top administrator cannot be added and deleted, and the security attributes are restricted to be modified by only top administrator.

- Operation administrator: An authorized administrator who has all the rights. He/she can employ all functions specified in the security management and can add and delete administrators except for top administrators as well as designate and change their roles.

- Read-only administrator: An administrator assigned by top administrator and operation administrator. Read-only administrator has a right to search the abnormal presence in the major processes in TOE, current status of pattern update, the number of collected files as of the search date, and the number of daily malicious code detections (within the recent week) through the administrator web page (GUI). He/she has also a right to search all audit data.

- Monitoring administrator: An administrator assigned by top administrator and operation administrator. Read-only administrator has a right to search the abnormal presence in the major processes in TOE, current status of pattern update, the number of collected files as of the search date, and the number of daily malicious code detections (within the recent week) through the administrator web page.

The specifications of the security management functions are as follows:

Modification of the authorized administrator setup

ZombieZERO Detector defines the roles of the administrators for authorized administrators (top administrator and operation administrator). The password at the time of administrator creation can be entered between 9-15 characters. ZombieZERO Detector has restriction rules about the security attributes of administrator account i.e., ID, password, access right, and lock or unlock, which are enforced by ZombieZERO Detector.

Please refer to "7.4.3 Password Verification" for more details of the password setup rules, which are divided into allowable characters, combination rules, and minimum/maximum lengths.

Authorized administrators can search, add, modify and delete administrator IDs and passwords that are allowed to access the TOE security management interface during the operation of TOE. The TOE performs identification and authentication of administrators based on the values predefined by authorized administrators. ZombieZERO Detector generates audit data when authorized administrators add, modify or delete the administrator ID list, or modify the security attributes of objects during operation.

Connection permission management

ZombieZERO Detector provides a function to manage IP address, administrator IDs and passwords that are allowed to access the security management interface (HTTPS/TLS 1.2).

Authorized administrators can search, add, modify and delete IP addresses that are allowed to access the TOE security management interface during operation of TOE. The TOE generates audit data when authorized administrators add, modify or delete the administrator IP addresses during operation.

Alarm setup

ZombieZERO Detector provides a function to set up the alarm criteria and method when authorized administrators violate the disk usage limit in the operational environment of ZombieZERO Detector. That is, ZombieZERO Detector performs an alarm function about the TOE operational environment based on the value set by authorized administrators.

ZombieZERO Detector sends an alarm through SMS or email designated by authorized administrators once the disk usage reaches the pre-defined limit.

ZombieZERO Detector generates audit data when authorized administrators modify the security attributes of alarm function, that is, alarm criteria, mail information and mobile phone numbers during operation.

Setup of Integrity inspection time

ZombieZERO Detector provides a function to update or inspect the data integrity by authorized administrators. When authorized administrators perform integrity update or inspection function, ZombieZERO Detector performs self-testing on the execution code and then notifies the results through the security management interface. If integrity-related violations are detected, authorized administrators may take a reaction such as ignoring the violation or initialization. Authorized administrators can set the time interval of self-testing. The TOE generates audit data when authorized administrators perform integrity inspection during operation.

Session timeout setup

Authorized administrators can set up the “time limit (session timeout: 5-10 min, 10 min default)” that is user session limit time through the security management interface. The default value is applied if authorized administrators do not set a specific value. Afterward, ZombieZERO Detector performs a user session termination function based on the above value, and generates audit data.

ZombieZERO Detector IP setup

Authorized administrators can set up an IP address of ZombieZERO Detector through the security management interface. The IP of ZombieZERO Detector is an IP address of the administrator web page (GUI) accessed by authorized administrators to perform the security management functions. Top administrator can set up the initial IP in the initial setup page provided by ZombieZERO Detector once the ZombieZERO Detector is installed normally. ZombieZERO Detector performs IP change function of ZombieZERO Detector and generates audit data when authorized administrators perform IP setup of ZombieZERO Detector during operation.

ZombieZERO Agent policy setup

Authorized administrators can set up the policy of ZombieZERO Agent through the security management interface. The policy of ZombieZERO Agent is a security policy to order the reaction that shall be taken

by ZombieZERO Agent at the time of malicious code detection in the ZombieZERO Detector. The policy types of ZombieZERO Agent are divided into three types: detection, block, and isolation. Once authorized administrators set up the policy of ZombieZERO Agent through the security management interface, ZombieZERO Detector sends the policy to the ZombieZERO Agent and generates audit data. The description of each of the security policies is explained in Section 7.7.3 Reaction.

Pattern management

When the update files about the following patterns owned by ZombieZERO Detector from the update server run by NPCore Inc. are received, the updates are applied after signature verification (RSA 2048) on the update files.

- List of malicious code hash values (black list)
- List of normal file hash values (white list)
- Yara rules
- Unauthorized IP/URL list (e.g., malicious bot distribution servers and C&C server)

It provide a function to manage (modification, deletion, and addition) of patterns through the security management interface for authorized administrators.

ZombieZERO Detector policy setup

Authorized administrators can set up the policies of ZombieZERO Detector through the security management interface. The security policies of ZombieZERO Detector are divided into six categories: “maximum collection size”, “collection protocol”, “package file handling”, “automatic blacklist handling, “automatic white list handling”, and “unknown file handling”.

- **Maximum collection size:** A function, by which authorized administrators can set up the maximum file size collected by ZombieZERO Detector during file collection through the security interface, is provided.
- **Collection protocol:** A function by which authorized administrators can initiate or suspend the collection of each of the following protocols: HTTP 1.1, SMTP, POP3, IMAP, and FTP by ZombieZERO Detector through the security interface, is provided.
- **Package file handling:** A function, by which authorized administrators can designate the number of decompressions through the security interface, is provided when ZombieZERO Detector collects compressed files.
- **Automatic black list handling (storing malicious file in the list of malicious file):** A function, by which authorized administrator can initiate or suspend actions of extracting hash values of files containing malicious code followed by storing them into the malicious file list through the security interface, is provided when the analysis result is detected as malicious by ZombieZERO Detector.
- **Automatic white list handling (normal files are stored in the normal file list):** A function, by which authorized administrator can initiate or suspend actions of extracting hash values of normal files followed by storing them into the normal file list through the security interface, is provided when the analysis results is detected as normal by ZombieZERO Detector.
- **Unknown file handling (handling of files whose analysis results cannot be categorized between**

malicious and normal detections.): When the analysis result was “unknown”, which cannot be classified either normal or malicious detections, a function, by which authorized administrators can register this in the black list by default and then revise the decision later, is provided.

7.5.2 TSF Data Management

ZombieZERO Detector provides a function to manage TSF data (security policy data) of TSF (security functions) to authorized administrators according to their roles (top administrator, operation administrator, read right administrator, and monitoring administrator). Top administrator and operation administrator provides a function to add, delete, and modify environment setup data and security policy data required to operate the TOE through the administrator web page (GUI). (FMT_MTD.1), (FMT_SMF.1), (FMT_SMR.1)

[Table 13] List of TSF Data Operations

Administrator role / TSF data	Top Administrator	Operation Administrator	Read right administrator	Monitoring administrator
Security audit query	Query	Query	Query	-
Administrator connection setup	Addition, modification, deletion	Addition, modification, deletion	-	
System setup	modification	modification	-	
Security path setup	modification	modification		
Initialization	modification	modification	-	-
ZombieZERO Agent policy setup	Query, addition, modification, deletion	Query, addition, modification, deletion	-	-
ZombieZERO Detector policy setup	modification	modification	-	-
Pattern management	Query, addition, modification, deletion	Query, addition, modification, deletion	-	-

7.6 TSF Protection

ZombieZERO Detector performs self-testing at the time of startup, and during operations periodically to demonstrate the correct operation of all TSFs. In addition, authorized administrators may perform the integrity inspection through the TOE security management interface. ZombieZERO Detector generates hash values of execution files required to operate ZombieZERO Detector, and performs self-testing by comparing the generated and stored hash values at the time of fist startup. If integrity-related violations are discovered, ZombieZERO Detector notifies this through the security management interface, and generates audit data.

ZombieZERO Agent performs self-testing at the time of TOE startup and periodically during normal operation. ZombieZERO Agent extract the registry setup values and performs self-testing by comparing the extracted and stored registry values at the time of first startup.

ZombieZERO Detector and ZombieZERO Agent record the audit results after self-testing. (FPT_TST.1(1)), (FPT_TST.1(2))

7.7 TOE Access

ZombieZERO Detector terminates the session if authorized administrator connects to the security management web page (GUI) and then does not take any action (e.g., policy setup or query) for a pre-determined time (5 to 10 min, 10 min default). Once the inactive time (5 to 10 min, 10 min default) of administrator is passed, the administrator's session is terminated. If no specific value for the inactive time of administrator is not set up, 10 min of default value is applied. ZombieZERO Detector performs a user session termination function based on the above value, and generates audit data. (FTA_SSL.3)

7.8 Malicious Code Detection and Block

7.8.1 Information Collection

The TOE performs block or isolation of malicious code in user PCs after detecting files that include malicious code to protect user PCs in the internal network. ZombieZERO Detector extracts and stores files

from the traffic introduced to the internal network from external networks to perform the security functions. (DSM_COL.1)

The information collected in ZombieZERO Detector is as follows:

Collected traffic type
ZombieZERO Detector receives protocols data of HTTP 1.1, IMAP, POP3, FTP, and SMTP Transmission from external to internal networks through the network TAP.

Collected file type
ZombieZERO Detector extracts executable PE files and document file extension (MS-Office, Hancom) files from the received Windows and stores them.

File collection type in user PCs
ZombieZERO Detector collects unauthorized exe files executed in ZombieZERO Agent during operations of ZombieZERO Detector and ZombieZERO Agent and stores them.

Analyzer Agent executes files Transmission from ZombieZERO Detector and records information of file actions, process actions, registry actions, network actions, and memory actions as a log while the file is executed and stores the log as a file format. (DSM_COL.2)
Executable file types in Analyzer Agent
Analyzer Agent shall execute files of the following file extensions: EXE, DOC, XLS, PPT, HWP, PDF, DLL, SWF, XLSM, XLTM, XLAM, DOCM, PPTM, JPG, JPEG, GIF, and JS shall be executed to record the file actions.

7.8.2 Information Analysis

ZombieZERO Detector performs the three-phase analysis based on the collected information: 1) signature comparison, 2) comparison between collected information and detection rules, and 3) comparison between behavior information of collected file and detection rules in sequence. In addition, each analysis phase performs whether malicious code is detected and if malicious code is detected, the next phase is not performed. If malicious code is not detected in the final phase step 3, the result of normal file output is produced. (DSM_INA.1)

Comparison between collected information and signature pattern
ZombieZERO Detector compares the collected file with the signature pattern to detect malicious code. If malicious code is not discovered in the signature pattern, the comparison through the detection rule is requested.

Comparison between collected information and detection rule
ZombieZERO Detector compares the file that has no malicious code discovered in the signature pattern with the detection rule to detect malicious code.

Comparison between file execution information and Yara rule
Malicious code is detected through the comparison between detection rule and file execution information collected through Analyzer Agent. If malicious code is not discovered by the comparison between file execution information and detection rule, normal file output is produced.

7.8.3 Reaction

ZombieZERO Detector sends the reaction policy (detection, block, or isolation) to ZombieZERO Agent when malicious code is detected in the collection information analysis phase. (DSM_RCT.1(1))

ZombieZERO Agent performs detection, block or isolation of malicious code based on the transferred policy from ZombieZERO Detector. (DSM_RCT.1(2))

ZombieZERO Agent ‘detection’ policy
ZombieZERO Agent sends a message of detection to ZombieZERO Detector after detecting the unauthorized file when the “detection” policy is received from ZombieZERO Detector.

ZombieZERO Agent ‘block’ policy
ZombieZERO Agent immediately blocks the file process that includes the malicious code when the “block” policy is received from ZombieZERO Detector.

ZombieZERO Agent ‘isolation’ policy
ZombieZERO Agent immediately blocks the file process that includes the malicious code when the “isolation” policy is received from ZombieZERO Detector, and isolates the file to the place designated by the developer.