OPSWAT.

MetaDefender Core & MetaDefender Kiosk

Security Target

Version 1.3

March 2021

Document prepared by

Lightship Security

www.lightshipsec.com

Document History

Version	Date	Author	Description
1.0	2 Nov, 2020	L Turner	Release for certification.
1.1	29 Jan, 2021	L Turner	Address certification comments.
1.2	17 Feb, 2021	L Turner	Address certification comments.
1.3	1 Mar, 2021	L Turner	Address certification comments.

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the OPSWAT MetaDefender Core & MetaDefender Kiosk Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 OPSWAT MetaDefender is a cybersecurity platform for detecting and preventing cybersecurity threats on multiple data channels. MetaDefender Core is a server component that provides centralized file analysis orchestration capabilities. MetaDefender Kiosk is a front-end component that is used as a media scanning workstation.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	MetaDefender Core v4.19.0 & MetaDefender Kiosk v4.4.5 Build: MetaDefender Core v4.19.0 & MetaDefender Kiosk v4.4.5.3699
Security Target	MetaDefender Core & MetaDefender Kiosk Security Target, v1.3

1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) EAL2+ (augmented with ALC_FLR.1)

b) CC version 3.1 Release 5

c) CC Part 2 extended

d) CC Part 3 conformant

1.4 Terminology

Table 2: Terminology

Term	Definition
AI/NGAV	Artificial Intelligence / Next Generation Anti-virus A combination of artificial intelligence, behavioural detection, machine learning algorithms, and exploit mitigation, to detect previously unknown threats.
CC	Common Criteria
CDR	Content Disarm and Reconstruction
CIDR	Classless Inter-Domain Routing, an IP addressing scheme
DLP	Data Loss Prevention

Term	Definition
EAL	Evaluation Assurance Level
PP	Protection Profile
TOE	Target of Evaluation
TSF	TOE Security Functionality

2 TOE Description

2.1 Type

4 The TOE is a file-based threat detection and prevention solution.

2.2 Usage

5 The TOE is typically deployed into secure environments that require all portable media to be scanned on entry and/or exit.

6 An example deployment of the TOE (enclosed in red) is shown in Figure 1. **Note:** It is not necessary that the TOE be deployed on an isolated network, nor is it necessary for the environment to include the Binary Armor or Secure File Transfer products.

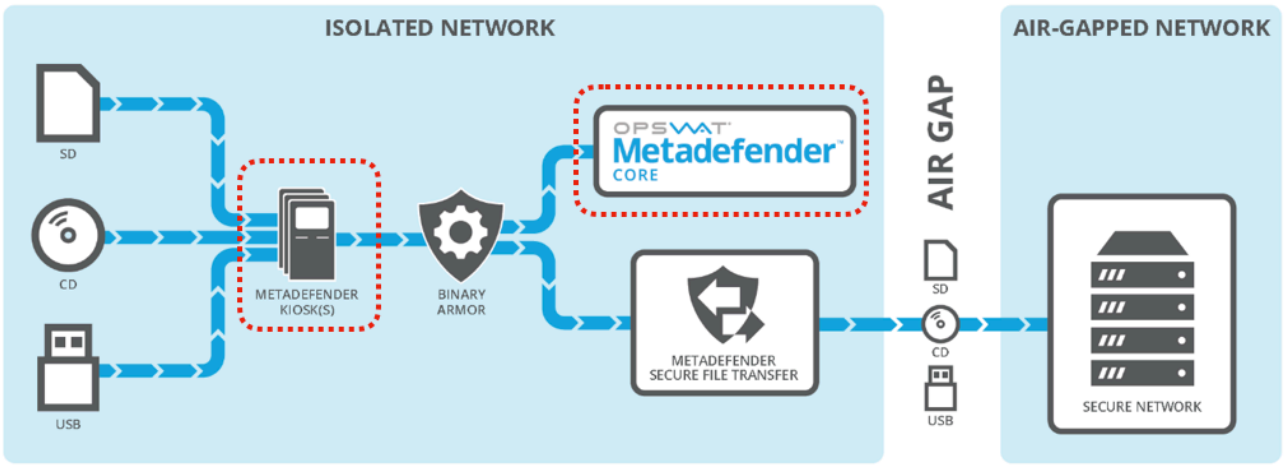


Figure 1: Example TOE deployment

2.2.1 MetaDefender Kiosk

7 Media such as USB devices, DVDs, card readers, SD cards, flash drives, or floppy disks, are scanned by MetaDefender Kiosk by inserting the media device into the appropriate drive. After the scan is complete, Kiosk generates a detailed report.

Figure 2: Kiosk Usage

8 The Kiosk has the following relevant usage characteristics:

a) **Management Console.** The MetaDefender Kiosk Management Console Web UI allows remote management via HTTPS.

b) **User Authentication.** MetaDefender supports scanning user authentication for audit and policy enforcement purposes.

2.2.2 MetaDefender Core

9 MetaDefender Kiosk uses MetaDefender Core to process files. MetaDefender Core has the following usage characteristics:

a) **REST API.** MetaDefender Core implements a REST API over HTTPS. All file processing (e.g. Kiosk or Web UI) occurs via this JSON-based interface.

b) **Management Console.** The MetaDefender Core Management Console Web UI allows remote management via HTTPS. Prior to authentication at the MetaDefender Core server's URL, the public file processing interface will be displayed. This page allows direct upload of files for processing (see Figure 3).

c) **File Processing.** MetaDefender Core has the following file processing capabilities:

i) Scanning with multiple anti-malware engines

ii) Deep Content Disarm and Reconstruction (CDR) / Data sanitization

iii) File-based vulnerability assessment

iv) Proactive Data Loss Prevention (DLP)

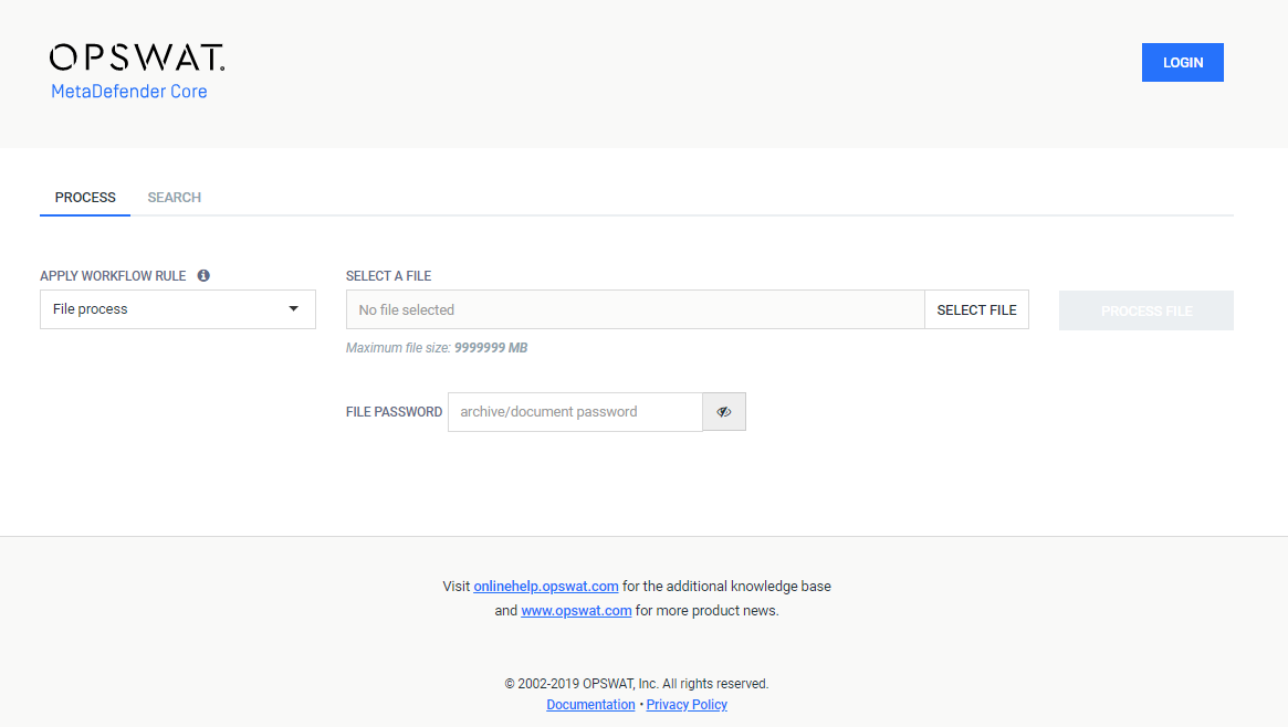


Figure 3: MetaDefender Core Web UI

2.3 Security Functions / Logical Scope

10 The TOE provides the following security functions:

a) **File Threat Analysis.** The TOE orchestrates the analysis of files for threats and generates associated scanning session reports. Based on scan results, files are handled according to administrator defined policies for 'Blocked Files' and 'Allowed Files'. Scan types include:

   i) **Scanning with multiple anti-malware engines.** File scanning in your environment (no data is shared outside) using over 30 anti-malware engines, including signature-based detection, AI/NGAV and also heuristic detection.

   ii) **Deep CDR / Data sanitization.** Remove active content from common types of document and image files by either converting the file format or removing hidden exploitable objects such as scripts and macros.

   iii) **File-based Vulnerability Assessment**. Ability to identify all known vulnerabilities in binaries (applications, patches, firmware updates) that might be used to exploit and compromise the end-user system once installed/deployed.

   iv) **Proactive DLP.** Detect, redact, watermark or block sensitive data in supported file types. Sensitive data may include credit card numbers, social security numbers or any specific data pattern using a regular expression.

b) **Protected Communications.** The TOE makes use of HTTPS/TLS to protect communication with remote administrators and between the Kiosk and Core.

c) **User Authentication Support.** The TOE supports authenticating users as follows:

   i) **Kiosk Scanning User.** Kiosk scanning users are authenticated using Windows Login. Guest users may also perform scans depending on the defined policy.

   ii) **Kiosk Management Console User.** Kiosk administrators are authenticated by means of a username and password against a local database.

   iii) **Core REST API / Management Console User.** Core users are authenticated by means of a username and password against a local database or Active Directory.

d) **Security Management.** The TOE enables secure management of its security functions, including enforcing role-based access control, generating security audit events and performing trusted software updates, including updates to engines and signatures, using digital signatures.

2.3.1 Physical Scope

11 The TOE includes the following software:

a) OPSWAT MetaDefender Core v4.19.0 (Windows version) with the modules and engines shown in Table 3[1].

b) OPSWAT MetaDefender Kiosk v4.4.5.3699

Table 3: Modules and Engines

Name	Type	Version
Aegis Lab	Metascan Engine	6.0-52

[1] Note that modules and engines are regularly updated. These were the versions installed during testing.

Name	Type	Version
Ahnlab	Metascan Engine	3.12.1.2 -457
Antiy	Metascan Engine	3.0.5.5-36
Avira	Metascan Engine	4.14.4-482
Bitdefender	Metascan Engine	3.0.1.219-80
ByteHero	Metascan Engine	1-32
ClamAV	Metascan Engine	0.102.4-524
Comodo	Metascan Engine	6.5.0.878-112
CrowdStrike	Metascan Engine	1.1.0-54
Cyren	Metascan Engine	6.2.0-64
Emsisoft	Metascan Engine	2018.04.0.1029-197
Eset	Metascan Engine	1462 (20150625)-55
Filseclab	Metascan Engine	1.0.2.2087-46
Huorong	Metascan Engine	67110146-18
Ikarus	Metascan Engine	5.4.6-464
K7	Metascan Engine	12.8.0.1-51
Kaspersky	Metascan Engine	8.3.2.4-51
Lavasoft	Metascan Engine	11.15-36
McAfee	Metascan Engine	6200-332
Nano AV	Metascan Engine	1.0.38.74417-27
Netgate	Metascan Engine	11.0.195.0-49
QuickHeal	Metascan Engine	13.00-66
RocketCyber	Metascan Engine	08_03_2020-100
Sophos	Metascan Engine	3.81.0-283
Symantec	Metascan Engine	7.9.1.12-240
Systweak	Metascan Engine	2.1.1000.10229-43

Name	Type	Version
Trend Micro	Metascan Engine	9.800.1009-56
TACHYON	Metascan Engine	2020.4.22.1-383
Trend Micro HouseCall	Metascan Engine	9.800.1009-43
VirIT Explorer	Metascan Engine	9.1.1-432
VirusBlokAda	Metascan Engine	3.12.16-22
Windows Defender ATP	Metascan Engine	1-151
XVirus Personal Guard	Metascan Engine	3.0.1.0-47
Zillya	Metascan Engine	1.2.0.7-66
Proactive DLP	Module	2.6.1-1611928554
Deep CDR	Module	5.11.1-6798
File Based Vulnerability Assessment	Module	4.2.416.0-117

2.3.2 Guidance Documents

12 The TOE includes the following guidance documents:

a) OPSWAT MetaDefender Core v4.19.0 User Guide (HTML), https://onlinehelp.opswat.com/corev4/v4.19.0.html

b) OPSWAT MetaDefender Kiosk v4 User Guide (PDF), https://onlinehelp.opswat.com/MetaDefender%20Kiosk%204.4.5.pdf

c) OPSWAT MetaDefender Core & MetaDefender Kiosk Common Criteria Guide (PDF), v1.1

2.3.3 Non-TOE Components

13 The TOE operates with the following components in the environment:

a) **MetaDefender Kiosk OS.** MetaDefender Kiosk requires a 64-bit Windows OS. The evaluated configuration assumes Windows 10.

b) **MetaDefender Kiosk Hardware.** MetaDefender Kiosk requires hardware that supports the above Windows OS and desired portable media peripherals. Hardware may be user supplied or purchased from OPSWAT. OPSWAT hardware options[2] are described at: https://www.opswat.com/products/metadefender/kiosk/kiosk-options

[2] OPSWAT Kiosks come with an image of the OS secured according to Defense Information Systems Agency (DISA) compliant configurations following Security Technical Implementation Guide (STIG) guidelines with Kiosk and Core installed per https://onlinehelp.opswat.com/kiosk/1.7_Kiosk_Secure_Image.html

c) **MetaDefender Core OS.** MetaDefender Core supports Windows and Unix based deployments. The evaluated configuration assumes Windows Server 2016.

d) **Network Environment.** Although the TOE can be deployed in a stand-alone non-networked environment, the evaluated configuration assumes a network environment that provides connectivity between the Core and Kiosk.

2.3.4 Excluded Functionality

14 The following security related functionality that is available in MetaDefender Core & MetaDefender Kiosk has not been evaluated:

a) Use with Vault Server

b) Email password recovery

c) Custom scanners

d) Yara rule sources

e) Cloud based scanning by 3rd party malware engines

f) Sending files to MetaDefender Cloud

g) Decryption / unlock of password protected files

h) Kiosk visitor management

3 Security Problem Definition

3.1 Threats

Table 4: Threats

Identifier	Description
T.FILE_THREAT	Attackers use file-based malware on portable media to compromise TOE protected IT resources.
T.DATA_LOSS	Attackers exfiltrate sensitive information, or users inadvertently transfer sensitive information between domains (e.g. high to low), on portable media.
T.MGMT	Attackers compromise the integrity of TOE security policies via TOE management interfaces.
T.COMMS	Attackers compromise the confidentiality or integrity of communication between TOE components or between the TOE and remote users.

3.2 Assumptions

Table 5: Assumptions

Identifier	Description
A.ADMIN	Administrators are trusted and follow guidance.
A.USER	Non-administrative users of the TOE are trusted and follow guidance.
A.PHYSICAL	TOE components are protected from unauthorized physical access.
A.TIME	The IT environment will provide a reliable time source.

3.3 Organizational Security Policies

Table 6: Organizational Security Policies

Identifier	Description
OSP.AUDIT	The TOE shall be capable of auditing the use of scanning and management functions.

4 Security Objectives

4.1 Objectives for the Operational Environment

Table 7: Security Objectives for the Operational Environment

Identifier	Description
OE.ADMIN	Administrators shall be trustworthy and follow guidance.
OE.USER	Non-administrative users of the TOE shall be trustworthy and follow guidance.
OE.PHYSICAL	TOE components shall be protected from unauthorized physical access.
OE.TIME	The IT environment will provide a reliable time source.

4.2 Objectives for the TOE

Table 8: Security Objectives

Identifier	Description
O.DETECT	The TOE shall enable detection of file-based threats and respond according to a defined policy.
O.DLP	The TOE shall detect sensitive data in submitted files and respond according to a defined policy.
O.ACCESS	The TOE shall prevent unauthorized access to management interfaces.
O.MGMT_AUDIT	The TOE shall audit usage of management interfaces.
O.KIOSK_AUDIT	The TOE shall audit the use of kiosk scanning functions.
O.COMMS	The TOE shall protect communication between TOE components and between the TOE and remote users.
O.UPDATE	The TOE shall authenticate software updates.

5 Security Requirements

5.1 Conventions

15 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

e) **Iteration.** Indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

5.2 Extended Components Definition

16 Table 9 identifies the extended classes, families and components which are incorporated into this ST, and a rationale for their creation.

Table 9: Extended Components

Title	Rationale
FPT_TUD: Trusted Update	Trusted update per Network Device Collaborative Protection Profile v2.1
Class FTR: Advanced Threat Prevention	The existing classes of the CC do not precisely address this class of security functionality, which is not specific to TSF data (FPT), User data (FDP), Security audit (FAU) or any other aspect covered by existing classes.
FTR_SCN: File Threat Scanning	The existing families of the CC do not address file-based threat scanning.
FTR_CDR: Content Disarm and Remove	The existing families of the CC do not address data sanitization / functionality to remove active content from files.
FTR_FVA: File-based Vulnerability Assessment	The existing families of the CC do not address functionality to identify known vulnerabilities in binaries.
FTR_DLP: Data Loss Prevention	The existing families of the CC do not address data loss prevention functions.
FTR_TAR: Threat Analysis Report	The existing families of the CC do not address production of threat reports.

5.2.1 Trusted Update (FPT_TUD)

5.2.1.1 Family Behavior

17 This family provides requirements that address trusted updates to the TSF.

5.2.1.2 Component Leveling

18 FPT_TUD.1 specifies requirements to update the TOE firmware and software, including the ability to verify the updates prior to installation.

5.2.1.3 Management: FPT_TUD.1

19 The following actions could be considered for the management functions in FMT:

a) None

5.2.1.4 Audit: FPT_TUD.1

20 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) None

FPT_TUD.1 Trusted Update

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic Operation

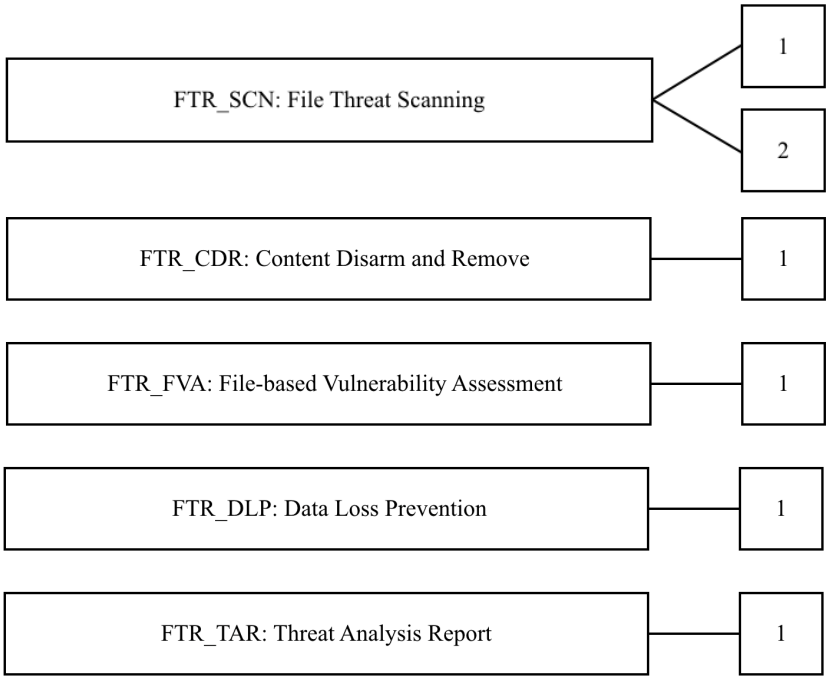
FPT_TUD.1.1 The TSF shall provide [assignment: *Administrators*] the ability to query the currently executing version of the TOE firmware/software and [selection: *the most recently installed version of the TOE firmware/software; no other TOE firmware/software version*].

FPT_TUD.1.2 The TSF shall provide [assignment: *Administrators*] the ability to manually initiate updates to TOE firmware/software and [selection: *support automatic checking for updates, support automatic updates, no other update mechanism*].

FPT_TUD.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: *digital signature mechanism, published hash*] prior to installing those updates.

5.2.2 Class FTR: Advanced Threat Prevention

21 This class contains families of functional requirements that relate to the prevention of advanced threats – sophisticated malware, data exfiltration or vulnerability-based attacks targeting sensitive data or IT assets. These threats are not necessarily against the TSF or user data held within a TOE but against organizational assets. Hence, organizations would deploy a TOE containing Advanced Threat Prevention functionality to defend against such threats.

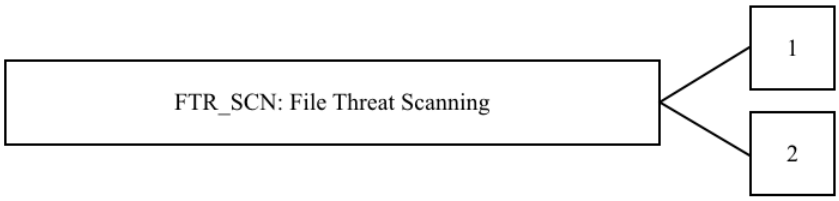


5.2.3 File Threat Scanning (FTR_SCN)

5.2.3.1 Family Behavior

22 This family provides requirements that address file-based threat scanning.

5.2.3.2 Component Leveling



23 FTR_SCN.1 specifies how files can be submitted for scanning.

24 FTR_SCN.2 specifies the scanning engines that are used for scanning.

5.2.3.3 Management: FTR_SCN.1

25 The following actions could be considered for the management functions in FMT:

a) Management of related policy configuration.

5.2.3.4 Audit: FTR_SCN.1

26 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Submission of files for scanning

FTR_SCN.1 Submission of Files for Scanning

Hierarchical to: No other components.

Dependencies: No other components.

FTR_SCN.1.1 The TSF shall support the following methods of file submission for scanning: [assignment: *list of the ways that files may be submitted for scanning]*.

FTR_SCN.2 Supported Scanning Engines

Hierarchical to: No other components.

Dependencies: FTR_SCN.1 Submission of Files for Scanning

FTR_SCN.2.1 The TSF shall [selection: *use 3rd party, implement its own*] scanning engines that perform the following types of threat scanning: [assignment: *list of supported scanning engines and associated scan types (this may be individual engines/scan types or identification of standards-based engine types that the TOE supports)*].

FTR_SCN.2.2 The scanning engines used by the TSF shall be [selection: *local to the TOE, cloud based,* [assignment*: other – describe where the scanning engines reside*]].

FTR_SCN.2.3 The TSF shall submit the following information and artifacts with a scan request: [assignment: *list of information and artifacts]*.

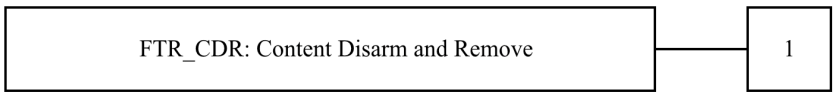
FTR_SCN.2.4 The TSF shall, at a minimum, receive the following information in the scan result: [assignment: *list of information*].

5.2.4 Content Disarm and Remove (FTR_CDR)

5.2.4.1 Family Behavior

27 This family provides requirements that address removing active content from files, such as embedded macros or other objects.

5.2.4.2 Component Leveling



28 FTR_CDR.1 specifies requirements for removing active content from files.

5.2.4.3 Management: FTR_CDR.1

29 The following actions could be considered for the management functions in FMT:

a) Management of related policy configuration

5.2.4.4 Audit: FTR_CDR.1

30 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) None

FTR_CDR.1 Content Disarm and Remove

Hierarchical to: No other components.

Dependencies: FTR_SCN.1 Submission of Files for Scanning

FTR_CDR.1.1 The TSF shall support removal of active content from the following types of files: [assignment: *list of supported file types*].

FTR_CDR.1.2 The TSF shall support removal of the following types of active content from files: [assignment: *list of active content*].

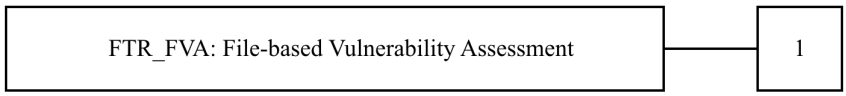
FTR_CDR.1.3 The TSF shall use the following methods to remove active content from files: [assignment: *list and describe methods used to remove active content*].

5.2.5 File-based Vulnerability Assessment (FTR_FVA)

5.2.5.1 Family Behavior

31 This family provides requirements that address identifying publicly known vulnerabilities in files.

5.2.5.2 Component Leveling



32 FTR_FVA.1 specifies requirements for identifying publicly known vulnerabilities in files.

5.2.5.3 Management: FTR_FVA.1

33 The following actions could be considered for the management functions in FMT:

a) Management of related policy configuration

5.2.5.4 Audit: FTR_FVA.1

34 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) None

FTR_FVA.1 File-based Vulnerability Assessment

Hierarchical to: No other components.

Dependencies: FTR_SCN.1 Submission of Files for Scanning

FTR_FVA.1.1 The TSF shall support vulnerability assessment of the following type files: [assignment: *supported file types*].

FTR_FVA.1.2 The TSF shall use the following reference sources for public vulnerabilities: [assignment: *vulnerability databases used*].

FTR_FVA.1.3 The TSF shall identify files of the supported file type that contain public vulnerabilities from the reference sources.

FTR_FVA.1.4 The TSF shall support the following actions when vulnerabilities are detected: [assignment: *actions*].

5.2.6 Data Loss Prevention (FTR_DLP)

5.2.6.1 Family Behavior

35 This family provides requirements that address techniques to prevent the loss of sensitive data such as credit card numbers, social security numbers or any specific data pattern. Techniques include:

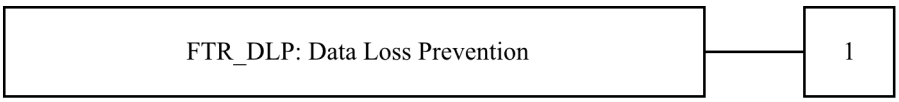
a) Detecting sensitive data

b) Redacting sensitive data

c) Watermarking documents containing sensitive data

d) Enforcing defined policies for files containing sensitive data

5.2.6.2 Component Leveling



36 FTR_DLP.1 specifies requirements to prevent the loss of sensitive data.

5.2.6.3 Management: FTR_DLP.1

37 The following actions could be considered for the management functions in FMT:

a) Management of related policy configuration

5.2.6.4 Audit: FTR_DLP.1

38 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) None

FTR_DLP.1 File-based Data Loss Prevention

Hierarchical to: No other components.

Dependencies: FTR_SCN.1 Submission of Files for Scanning

FTR_DLP.1.1 The TSF shall support applying data loss prevention techniques to the following types of files: [assignment: *supported file types*].

FTR_DLP.1.2 The TSF shall be able to identify the following types of sensitive data in files: [assignment: *types of sensitive data*].

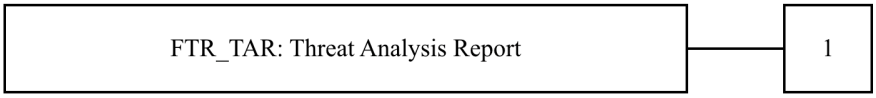
FTR_DLP.1.3 The TSF shall support the following actions when sensitive data is detected: [assignment: *actions*].

5.2.7 Threat Analysis Report (FTR_TAR)

5.2.7.1 Family Behavior

39 This family provides requirements that address production of threat reports.

5.2.7.2 Component Leveling



40 FTR_TAR.1 specifies requirements for production of threat reports.

5.2.7.3 Management: FTR_TAR.1

41 The following actions could be considered for the management functions in FMT:

a) Management of related policies

5.2.7.4 Audit: FTR_TAR.1

42 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) None

FTR_TAR.1 Threat Analysis Report

Hierarchical to: No other components.

Dependencies: FTR_SCN.2 Supported Scanning Engines, or
FTR_CDR.1 Content Disarm and Remove, or
FTR_FVA.1 File-based Vulnerability Assessment, or
FTR_DLP.1 File-based Data Loss Prevention
FMT_SMR.1 Security Roles

FTR_TAR.1.1 The TSF shall support generation of the following reports: [assignment: *list of reports*].

FTR_TAR.1.2 The TSF reports shall contain the following information: [assignment: *for each report, list the information contained in the report*].

FTR_TAR.1.3 The TSF shall allow the following roles to view the reports: [assignment: *roles*].

FTR_TAR.1.4 The TSF shall [selection: *not store reports, store reports as follows:* [assignment: *describe how reports are stored and the rules for retaining reports*]].

5.3 Functional Requirements

Table 10: Summary of SFRs

Requirement	Title
FTR_SCN.1	Submission of Files for Scanning
FTR_SCN.2	Supported Scanning Engines
FTR_CDR.1	Content Disarm and Remove
FTR_FVA.1	File-based Vulnerability Assessment
FTR_DLP.1	File-based Data Loss Prevention
FTR_TAR.1	Threat Analysis Report
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FCS_COP.1	Cryptographic Operation
FIA_UAU.1	Timing of authentication
FIA_UAU.7	Protected authentication feedback
FIA_UID.1	Timing of identification
FMT_MOF.1	Management of security functions behaviour
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security roles
FPT_ITT.1	Basic internal TSF data transfer protection
FPT_TUD.1	Trusted Update
FTP_TRP.1	Trusted path

5.3.1 Advanced Threat Protection (FTR)

FTR_SCN.1 Submission of Files for Scanning

Hierarchical to: No other components.

Dependencies: No other components.

FTR_SCN.1.1 The TSF shall support the following methods of file submission for scanning: [

- *user presentation of portable media at the Kiosk, and*
- *upload of files at the Core Management Console*].

FTR_SCN.2 Supported Scanning Engines

Hierarchical to: No other components

Dependencies: FTR_SCN.1 Submission of Files for Scanning

FTR_SCN.2.1 The TSF shall [use 3rd party] scanning engines that perform the following types of threat scanning: [*anti-malware scanning engines and scan types listed in the table below. Supported scan types:*

- ***Signature.*** *Signature-based malware scanning.*
- ***Heuristics.*** *Heuristics-based malware scanning.*
- ***PUA/PUP.*** *Scan for Potential Unwanted Applications (PUA) / Potentially Unwanted Program (PUP)*
- ***AI/ML.*** *Artificial Intelligence (AI) / Machine Learning (ML) supported malware scanning*]

	Scan Types (X = enabled by default, O = supported)			
Engine	**Signature**	**Heuristics**	**PUA/PUP**	**AI/ML**
Aegis Lab	X			
AhnLab	X		O	
Antiy	X	O		
Avira	X	X	X	O
Bitdefender	X	X		
ByteHero		X		
ClamAV	X	X	O	
Comodo	X			
CrowdStrike				X
Cyren	X		O	
Emsisoft	X	O		
Eset	X	X	X	
Filseclab	X			

	Scan Types (X = enabled by default, O = supported)			
Engine	**Signature**	**Heuristics**	**PUA/PUP**	**AI/ML**
Huorong	X			
Ikarus	X			
K7	X	O		
Kaspersky	X	X		X
Kicom AV				
Lavasoft	X	X		
McAfee	X	X	O	
Microsoft Security Essentials	X			
Windows Defender	X	X		X
NANO	X	X		
Netgate	X			
Tachyon	X	X		
Quick Heal	X	O	O	
RocketCyber				X
Sophos	X	O	O	
SparkCognition				X
Symantec	X	X	O	
Systweak	X			
Total Defense	X	X		
Trend Micro	X			
Trend Micro HouseCall	X			
VirIT	X		X	

	Scan Types (X = enabled by default, O = supported)			
Engine	**Signature**	**Heuristics**	**PUA/PUP**	**AI/ML**
VirIT ML				X
Virus Blokada	X			
Xvirus	X			
Zillya	X	O		

FTR_SCN.2.2 The scanning engines used by the TSF shall be [local to the TOE].

FTR_SCN.2.3 The TSF shall submit the following information and artifacts with a scan request: [*file and filename]*.

FTR_SCN.2.4 The TSF shall, at a minimum, receive the following information in the scan result: [

- *Threat name (if any)*
- *Scan result (malicious/suspicious/no threat found)*
- *Analysis Time in milliseconds*].

FTR_CDR.1 Content Disarm and Remove

Hierarchical to: No other components.

Dependencies: FTR_SCN.1 Submission of Files for Scanning

FTR_CDR.1.1 The TSF shall support removal of active content from the following types of files: [*file types listed in the table below*].

FTR_CDR.1.2 The TSF shall support removal of the following types of active content from files: [*content types listed in the table below*].

File Type	Content Type
doc	Macro
docm	OLE Objects:
docx	* Attachment
dot	* Embedded binary file
dotm	* Crafted embedded multimedia
dotx	* Script enabled ActiveX Controls
xls	Hyperlink
xlsm	Crafted images
xlsx	Embedded font (not supported for vsdx, vsdm)

File Type	Content Type
xlsb xlt xltx xltm ppt pptm pptx ppsx pps pot potx potm vsdx vsdm vsdx vssx vstx vsdm vssm vstm	Hidden text Comment Revision Metadata External media objects Timing node (pptx only) Mouse-Over Hyperlink/Click Hyperlink (pptx only) External image (docx only) Chart (not supported for xlsx)
rtf	Embedded object Suspicious Drawing object Embedded HTML Metadata
csv	Formula injection
htm/html	Images Embedded Objects Embedded Java applets Href Metadata
pdf	Hyperlink Actions/JavaScript Annotation

File Type	Content Type
	Attachments Multimedia Objects Images Embedded font Form fields (edit form, check box..) DTD Metadata
odt	Macro Embedded Object Hyperlink Images Embedded Font Metadata External image
jtd	Macro
jtdc	Hyperlink Embedded Objects Images Font Document View Styles
hwp	Embedded Objects * Flash Files * RTF * PCT Images Macro Hyperlinks
show	Embedded Objects Images Hyperlinks
cell	Embedded Objects Images

File Type	Content Type
	Macro Hyperlinks
ics	Attachment Hyperlinks
xml	XML bomb / oversized payload Recursive payload Cdata injection XML injection VB Macro Script
mp3	Metadata EOF Frame ID3 tag
wav	Metadata
svg	Javascript Cdata injection XML bomb
dwg jpg bmp png tiff	Macro Abnormal content Embedded malicious code: * HTML * PHP * Javascript * exploit code Metadata WMF/EMF only: * None standard EOF record * exploit codes

FTR_CDR.1.3 The TSF shall use the following methods to remove active content from files: [

- *rebuild the file without the active content, or*
- *convert the file to a different file type*].

FTR_FVA.1 File-based Vulnerability Assessment

Hierarchical to: No other components.

Dependencies: FTR_SCN.1 Submission of Files for Scanning

FTR_FVA.1.1 The TSF shall support vulnerability assessment of the following type files: [*exe, dll, sys, msi, cab, dmg, zip, tar, gz, bz2, bin*].

FTR_FVA.1.2 The TSF shall use the following reference sources for public vulnerabilities: [*National Vulnerability Database, Microsoft Security Update Guide*].

FTR_FVA.1.3 The TSF shall identify files of the supported file type that contain public vulnerabilities from the reference sources.

FTR_FVA.1.4 The TSF shall support the following actions when vulnerabilities are detected: [*report, block, allow*].

FTR_DLP.1 File-based Data Loss Prevention

Hierarchical to: No other components.

Dependencies: FTR_SCN.1 Submission of Files for Scanning

FTR_DLP.1.1 The TSF shall support applying data loss prevention techniques to the following types of files: [*Microsoft Offices (DOC, DOCX, XLS, XLSX, PPT, PPTX), OpenOffice (ODT, ODS, ODP), Adobe PDF, Text base (TXT, HTML, CSV, XML), Emails (EML, MSG), Images*].

FTR_DLP.1.2 The TSF shall be able to identify the following types of sensitive data in files: [

- *credit card numbers*
- *social insurance numbers*
- *Ipv4 address*
- *CIDR range*
- *regular expression data pattern*].

FTR_DLP.1.3 The TSF shall support the following actions when sensitive data is detected: [

- *block file*
- *attempt to delete the file from the original media*
- *report the type of sensitive data that was identified*].

FTR_TAR.1 Threat Analysis Report

Hierarchical to: No other components.

Dependencies: FTR_SCN.2 Supported Scanning Engines, and/or
FTR_CDR.1 Content Disarm and Remove, and/or
FTR_FVA.1 File-based Vulnerability Assessment, and/or

FTR_DLP.1 File-based Data Loss Prevention
FMT_SMR.1 Security Roles

FTR_TAR.1.1 The TSF shall support generation of the following reports: [*Kiosk Session Summary, Kiosk Scan Results*].

FTR_TAR.1.2 The TSF reports shall contain the following information: [

- *Kiosk Session Summary:*
  - *Number of Files Processed*
  - *Number, list and details of allowed files*
  - *Number, list and details of blocked files,*
  - *For each blocked file:*
    - *Details of threats found (malware scanning)*
    - *Details of sensitive data found (DLP scanning)*
    - *Details of potential vulnerabilities found (vulnerability assessment scanning)*
    - *Details of sanitized files (CDR scanning)*
    - *Details of password protected file*
    - *Details of archive file (too deep, too many files)*
- *Kiosk Scan Results*
  - *User ID*
  - *Profile (i.e. scan profile)*
  - *Session ID*
  - *Processing Time*
  - *Device Information (i.e. media scanned)*
  - *File Processing Details*
  - *Blocked Actions Taken*
  - *Allows Actions Taken*
  - *Detailed Scan Results*

].

FTR_TAR.1.3 The TSF shall allow the following roles to view the reports: [

Roles assigned with permissions: Overall Results Only, Per Engine Results, Full Details].

Application Note: Roles with Overall Results Only and Per Engine Results permissions may only view a subset of report information.

FTR_TAR.1.4 The TSF shall [store reports as follows*:* [*scan results are stored according to an administrator defined retention policy and reports are dynamically generated from stored scan results*]].

5.3.2 Security Audit (FAU)

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) [*Auditable events listed in the table below*].

Event	Additional Details
Log in to the Kiosk Management Console	-
User log in to the Kiosk application	-
Kiosk media insert \ removal	-
Core Scanning Request	Client IP address
Log in to the Core Management Console	-
User Management Operations	Operation Details

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, [*additional details specified in the above table*].

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.

5.3.3 Cryptographic Support (FCS)

FCS_COP.1 Cryptographic Operation

Hierarchical to: No other components.

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

FCS_COP.1.1 The TSF shall perform [*cryptographic operations shown in the table below*] in accordance with a specified cryptographic algorithm [*shown in the table below*] and cryptographic key sizes [*shown in the table below*] that meet the following: [*standards shown in the table below*].

Operation	Algorithm	Key Size	Standards	CAVP
Symmetric encryption and decryption	AES-GCM	128 256	ISO 18033-3 ISO 19772	C1903 C1904
Key exchange	ECDHE	P-256	NIST SP 800-56A	
Message digest	SHA2-256 SHA2-384	N/A	ISO/IEC 10118-3:2004	
Message authentication	HMAC-SHA2-256 HMAC-SHA2-384	256 384	ISO/IEC 9797-2:2011	
Signature Generation and Verification	RSA	2048 3072	FIPS 186-4	

5.3.4 Identification and Authentication (FIA)

FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1 The TSF shall allow [*Kiosk: presentation of files for scanning and view session reports (if guest submission is configured), Core: File Submission for Scanning*] on behalf of the user to be performed before the user is authenticated.

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

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 [*bullets*] to the user while the authentication is in progress.

FIA_UID.1 Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1 The TSF shall allow [*actions per FAU_UAU.1.1*] on behalf of the user to be performed before the user is identified.

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

5.3.5 Security Management (FMT)

FMT_MOF.1 Management of security functions behaviour

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 [determine the behaviour of, disable, enable, modify the behaviour of] the functions [*functions in the below table*] to [*roles in the below table*].

Role	Functions	Permissions
Kiosk		
Administrator	All	All
Auditor	Dashboard	Determine the behaviour of
	Logging History	Determine the behaviour of
Core		
Admin	All	All
Security Admin	All except User Management, Data Retention and Licensing	All

Role	Functions	Permissions
Security Auditor	All	Determine the behaviour of
Help Desk	Inventory and Workflow Configuration	Determine the behaviour of

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: [

Kiosk Management Console:

- *View Dashboard*
- *Configure TSF*

Core Management Console:

- *View Dashboard*
- *Configure TSF*

].

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 [*roles identified in FMT_MOF.1.1*].

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

5.3.6 Protection of the TSF (FPT)

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

Dependencies: No dependencies.

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

FPT_TUD.1 Trusted Update

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic Operation

FPT_TUD.1.1 The TSF shall provide [*all roles*] the ability to query the currently executing version of the TOE firmware/software and [no other TOE firmware/software version].

FPT_TUD.1.2 The TSF shall provide [*Core Admin*] the ability to manually initiate updates to TOE firmware/software and [support automatic updates].

FPT_TUD.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature mechanism] prior to installing those updates.

5.3.7 Trusted Path/Channels (FTP)

FTP_TRP.1 Trusted path

Hierarchical to: No other components.

Dependencies: No dependencies.

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

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

FTP_TRP.1.3 The TSF shall require the use of the trusted path for *MetaDefender Core WebUI/REST API and Kiosk Management Console WebUI.*

5.4 Assurance Requirements

43 The TOE security assurance requirements are summarized in Table 11 commensurate with EAL2+ (augmented with ALC_FLR.1).

Table 11: Assurance Requirements

Assurance Class	Components	Description
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 a CM System
	ALC_CMS.2	Parts of the TOE CM Coverage
	ALC_DEL.1	Delivery Procedures
	ALC_FLR.1	Basic Flaw Remediation
Security Target Evaluation	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.2	Security Objectives
	ASE_REQ.2	Derived Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Tests	ATE_COV.1	Evidence of Coverage
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent Testing – sample
Vulnerability Assessment	AVA_VAN.2	Vulnerability Analysis

6 TOE Summary Specification

6.1 File Threat Analysis

44 This security function implements the SFRs shown in Table 12.

Table 12: File Threat Analysis SFRs

Requirement	Title
FTR_SCN.1	Submission of Files for Scanning
FTR_SCN.2	Supported Scanning Engines
FTR_CDR.1	Content Disarm and Remove
FTR_FVA.1	File-based Vulnerability Assessment
FTR_DLP.1	File-based Data Loss Prevention
FTR_TAR.1	Threat Analysis Report
FMT_SMF.1	Specification of Management Functions

45 The TOE orchestrates the scanning and analysis of files for threats and generates associated session reports. According to administrator defined workflow/policies, the file analysis goes through multiple phases to detect if the file is either malicious, contains sensitive or vulnerable data and if it's safe to be consumed by the end-user.

46 MetaDefender Kiosk will submit all the selected files to MetaDefender Core for analysis. The submission happens as follows:

a) MetaDefender Kiosk will call MetaDefender Core's REST API to upload the file (POST / file)

b) MetaDefender Core will respond with a unique identifier for that file (data_id)

c) Multiple files will be submitted in parallel, until MetaDefender Core's queue is full

d) MetaDefender Core executes a multi-stage analysis flow for each submitted file

e) MetaDefender Kiosk will check periodically to see if the analysis is complete (short polling), using the previously received data_id

f) MetaDefender Kiosk will retrieve results and perform the associated file handling actions according to the administrator defined policy

g) MetaDefender Kiosk will present the summary report to the user

47 The TSF relevant technologies are described in the following sections.

6.1.1 Scanning with multiple anti-malware engines

48 MetaDefender Core offers the ability to license different packages for the multi-scanning module (called Metascan). Depending on the license key, the licensed third-party analysis engines are downloaded and deployed within Metascan.

49 This model ensures the end-customer can analyze all the files in their environment, with no data being submitted to either OPSWAT or the engine vendors for analysis. This guarantees that the solution works in online, offline and even air-gapped environments.

50 Once a file is submitted, the file is passed to all licensed engines for analysis, making use of parallel processing to ensure the highest throughput. When all the engines have completed the analysis, the result is provided back to MetaDefender Core.

6.1.2 Deep CDR / Data sanitization

51 Deep CDR allows user to sanitize productivity documents, by removing embedded active objects that might drive a malicious behavior (macros, OLE objects, ActiveX controls, etc. In Office docs).

52 Workflow configuration will allow the administrator to define, at file type level, which type will be sanitized and the method to use for sanitization, either:

a) CDR method (rebuild the file without the active content)

b) Filetype conversion method (convert a file to a different filetype, which will break the active content, but will also change the usability of the file, e.g. Excel file to PDF).

53 The supported file types and content are listed at FTR_CDR.1.2.

6.1.3 File-based Vulnerability Assessment

54 In case the submitted file is an application file (installer, patch, firmware update, etc.), the File-based Vulnerability Assessment Engine will map the file to its known vulnerabilities.

55 OPSWAT maintains a repository of known applications (and files belonging to those applications) and performs matching at file level to known vulnerabilities for those applications based on the following vulnerability databases:

a) National Vulnerability Database
https://nvd.nist.gov/

b) Microsoft Security Update Guide
https://portal.msrc.microsoft.com/en-us/security-guidance

6.1.4 Proactive DLP

56 The TOE supports applying data loss prevention techniques to

a) Microsoft Office files (DOC, DOCX, XLS, XLSX, PPT, PPTX)

b) OpenOffice files (ODT, ODS, ODP),

c) Adobe PDF,

d) Text base files (TXT, HTML, CSV, XML),

e) Email files (EML, MSG)

f) Images (TOE uses optical character recognition in this case)

57 The TOE is able to identify the following types of sensitive data in files based on the administrator defined policy:

a) credit card numbers

b) social insurance numbers

c) IPv4 address

d) CIDR range

e) regular expression data pattern

58 MetaDefender Core performs the analysis and will identify the sensitive data.

59 Kiosk will enforce the file handling policy to either:

a) block the file

b) attempt to delete the file from the original media

c) report the type of sensitive data that was identified

6.1.5 File Handling

60 The Kiosk enforces file handing actions for processed files.

61 Handling actions for Blocked files:

a) **No action.** Report only.

b) **Stop processing.** The Kiosk session to stop processing immediately after the first blocked file is found. Kiosk will alert the user that a blocked file was found and go directly to the session summary after the user has acknowledged the message.

c) **Remove file.** Blocked file will be removed from the original media and optionally quarantined.

d) **Sanitized file handling.** If a file has been sanitized, the original file may either be replaced by the sanitized file, or the sanitized file will be copied to the original media and the original file will be left untouched.

e) **Copy to.** Copy blocked files to specified locations. The original file may optionally be deleted.

62 Handling actions for allowed files include:

a) **No action.** Report only.

b) **Wipe and copy to original media.** Will copy allowed files back to the original media after the original media has been formatted.

c) **Sanitized file handling.** If a file has been sanitized (i.e. via CDR), the original file may either be replaced by the sanitized file, or the sanitized file will be copied to the original media and the original file will be left untouched.

d) **Copy to.** Copy allowed files to specified locations. The original file may optionally be deleted.

6.1.6 Reporting

63 After media has been processed, the session results appear. If any file was not processed by MetaDefender, a warning will pop up indicating that not all files were processed.

64 The session results include whether processing was completed or aborted, the number of files allowed and blocked and the total number of files processed.

65 The session result page includes the following buttons:

a) **Allowed**: If allowed files are found, then the Allowed count will appear. Click this button to go to the Allowed file summary screen.

b) **Blocked**: If blocked files are found, then the Blocked count will appear. Click this button to go to the Blocked file summary screen.

c) **Copy & Print**: Clicking this button will begin the file transfer process to any destination configured. If printing is enabled, the session results will be printed to the default printer.

66 The Blocked File Details screen displays the blocked files detected by MetaDefender Kiosk during processing. The user may click a blocked file to view more details.

6.2 Protected Communications

67 This security function implements the SFRs shown in Table 12.

Table 13: Protected Communications SFRs

Requirement	Title
FCS_COP.1	Cryptographic Operation
FPT_ITT.1	Basic internal TSF data transfer protection
FTP_TRP.1	Trusted path

68 The TOE protects communication using TLS as follows:

a) **MetaDefender Core.** Implements a web server (nginx) that requires HTTPS for the REST API and Management Console Web GUI.

b) **MetaDefender Kiosk.** Implements a web server (nginx) that requires HTTPS for access to the Management Console Web GUI. In addition, the Kiosk makes use of the Core's REST API by submitting request over HTTPS.

69 In all cases, the underlying TLS implementation is provided by OpenSSL. In the evaluated configuration, the TLS implementation has the following characteristics:

a) TLS 1.2 is supported

b) Supported ciphers:

i) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

ii) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

6.3 User Authentication

70 This security function implements the SFRs shown in Table 12.

Table 14: User Authentication SFRs

Requirement	Title
FIA_UAU.1	Timing of authentication
FIA_UAU.7	Protected authentication feedback
FIA_UID.1	Timing of identification

6.3.1 Kiosk Scanning Users

71 In the evaluated configuration, Kiosk scanning users are authenticated using Windows Login (i.e. the TOE invokes Windows Login). TOE administrators can choose to allow guest users, and whether to restrict the users by domain. If selected, only users on the same domain as the system are allowed to use MetaDefender Kiosk. If this is not selected, users will be able to enter authentication information for users on any domain as well as local system users.

6.3.2 Kiosk Management Console User

72 Kiosk administrates are authenticated by means of a username and password against a local database.

6.3.3 Core REST API / Management Console User

73 Core users are authenticated by means of a username and password against a local database or Active Directory.

6.4 Security Management

74 This security function implements the SFRs shown in Table 12.

Table 15: Security Management SFRs

Requirement	Title
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FCS_COP.1	Cryptographic Operation
FMT_MOF.1	Management of security functions behaviour
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security roles
FPT_TUD.1	Trusted Update

6.4.1 Kiosk Management Console

75 The Kiosk Management Console provides the following management functions:

a) Create / manage users

b) View the Dashboard - the first page that is seen when logging in to the MetaDefender Kiosk Management Console. This page provides a summary of all of the files that have been processed by MetaDefender Kiosk.

c) Configuration - the configuration pages allow administrators to configure all MetaDefender Kiosk settings that apply to all users of MetaDefender Kiosk.

76 Kiosk Management Console users are assigned to roles as defined at FMT_SMR.1. The TOE will enforce access control in accordance with the privileges assigned to each role.

6.4.2 Core Management Console

77 The Core Management Console provides the following management functions:

a) Create / manage users

b) View Dashboard - gives a general overview of MetaDefender Core status.

c) Configuration - the configuration pages allow administrators to configure all MetaDefender Core settings.

78 Core Management Console users are assigned to roles as defined at FMT_SMR.1. The TOE will enforce access control in accordance with the privileges assigned to each role.

6.4.3 Security Audit

79 The TOE generates audit logs and stores them locally. Each TOE component (Kiosk and Core) maintains its own audit log. The audit events and details are described at FAU_GEN.1.

80 Each audit event includes the date and time of the event, type of event, subject identity (if applicable), and the outcome of the event.

6.4.4 Trusted Update

81 MetaDefender Core implements trusted updates for signature files and installed 3rd party engines as follows:

a) All updates are digitally signed with the OPSWAT code signing private key (RSA / SHA2-256)

b) Updates are fetched from OPSWAT cloud infrastructure

c) Core uses the OPSWAT code signing public key to verify the digital signature prior to installing

82 Updates to the Core and Kiosk software are performed manually and are verified by the Microsoft Authenticode mechanism.

7 Rationale

7.1 Security Objectives Rationale

83 Table 16 provides a coverage mapping between security objectives, threats, OSPs and assumptions.

Table 16: Security Objectives Mapping

	T.FILE_THREAT	T.DATA_LOSS	T.MGMT	T.COMMS	A.ADMIN	A.USER	A.PHYSICAL	A.TIME	OSP.AUDIT
O.DETECT	X								
O.DLP		X							
O.ACCESS			X						
O.MGMT_AUDIT									X
O.KIOSK_AUDIT									X
O.COMMS				X					
O.UPDATE			X						
OE.ADMIN					X				
OE.USER						X			
OE.PHYSICAL							X		
OE.TIME								X	

84 Table 17 provides the justification to show that the security objectives are suitable to address the security problem.

Table 17: Suitability of Security Objectives

Element	Justification
T.FILE_THREAT	**O.DETECT.** Mitigates this threat by detected and responding to file-based threats.

Element	Justification
T.DATA_LOSS	**O.DLP.** Mitigates this threat by detecting sensitive data in scanned media and responding according to a defined policy.
T.MGMT	**O.ACCESS.** Mitigates this threat by preventing unauthorized access to management interfaces. **O.UPDATE.** Mitigates this threat by authenticating software updates (that are received via management interfaces).
T.COMMS	**O.COMMS.** Mitigates this threat by requiring protected communication between TOE components and between the TOE and remote users.
OSP.AUDIT	**O.MGMT_AUDIT.** Satisfies this OSP by requiring audit of the usage of management interfaces. **O.KIOSK_AUDIT.** Satisfies this OSP by requiring audit of the usage of Kiosk scanning functions.
A.ADMIN	**OE.ADMIN.** Restates the assumption as an environmental objective.
A.USER	**OE.USER.** Restates the assumption as an environmental objective.
A.PHYSICAL	**OE.PHYSICAL.** Restates the assumption as an environmental objective.
A.TIME	**OE.TIME.** Restates the assumption as an environmental objective.

7.2 Security Requirements Rationale

7.2.1 SAR Rationale

85 EAL2 was chosen to provide a level of assurance that is consistent with good commercial practices with the addition of ALC_FLR.1 to provide assurance that any identified security flaws will be addressed.

7.2.2 SFR Dependency Analysis

Table 18: SFR Dependency Analysis

SFR	Dependencies	Met / Rationale if not met
FTR_SCN.1	None	-
FTR_SCN.2	FTR_SCN.1	Met
FTR_CDR.1	FTR_SCN.1	Met

SFR	Dependencies	Met / Rationale if not met
FTR_FVA.1	FTR_SCN.1	Met
FTR_DLP.1	FTR_SCN.1	Met
FTR_TAR.1	FTR_SCN.2	Met
	FTR_CDR.1	Met
	FTR_FVA.1	Met
	FTR_DLP.1	Met
	FMT_SMR.1	Met
FAU_GEN.1	FPT_STM.1	Not met. Per OE.TIME, the environment provides reliable time.
FAU_GEN.2	FAU_GEN.1	Met
	FIA_UID.1	Met
FCS_COP.1	FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1	Not met. Per Canadian Common Criteria Scheme instructions.
	FCS_CKM.4	Not met. Per Canadian Common Criteria Scheme instructions.
FIA_UAU.1	FIA_UID.1	Met
FIA_UAU.7	FIA_UAU.1	Met
FIA_UID.1	None	-
FMT_MOF.1	FMT_SMR.1	Met
	FMT_SMF.1	Met
FMT_SMF.1	None	-
FMT_SMR.1	FIA_UID.1	Met
FPT_ITT.1	None	-
FPT_TUD.1	FCS_COP.1	Met
FTP_TRP.1	None	-

7.2.3 SFR Rationale

Table 19: Security Requirements Mapping

	O.DETECT	O.DLP	O.ACCESS	O.MGMT_AUDIT	O.KIOSK_AUDIT	O.COMMS	O.UPDATE
FTR_SCN.1	X	X					
FTR_SCN.2	X						
FTR_CDR.1	X						
FTR_FVA.1	X						
FTR_DLP.1		X					
FTR_TAR.1	X	X					
FAU_GEN.1				X	X		
FAU_GEN.2				X	X		
FCS_COP.1						X	X
FIA_UAU.1			X				
FIA_UAU.7			X				
FIA_UID.1			X				
FMT_MOF.1			X				
FMT_SMF.1	X	X	X				
FMT_SMR.1			X				
FPT_ITT.1						X	
FPT_TUD.1							X
FTP_TRP.1						X	

Table 20: Suitability of SFRs

Objectives	SFR supports the objective by requiring:
O.DETECT	FTR_SCN.1 – submission of files for scanning FTR_SCN.2 – scanning engines to detect malware FTR_CDR.1 – removal of potentially malicious content from files FTR_FVA.1 – detection of application files containing known vulnerabilities FTR_TAR.1 – reporting scanning results FMT_SMF.1 – specification of file handling / blocking policies Together the SFRs will result in the detection of file-based threats and response according to policy.
O.DLP	FTR_SCN.1 – submission of files for scanning FTR_DLP.1 – detection of sensitive data within files FTR_TAR.1 – reporting scanning results FMT_SMF.1 – specification of file handling / blocking policies Together the SFRs will result in the detection sensitive data in files and response according to policy.
O.ACCESS	FIA_UAU.7 – protection of authentication feedback FIA_UAU.1 – authentication of users FIA_UID.1 – identification of users FMT_MOF.1 – management of security functions behavior FMT_SMF.1 – specification of management functions FMT_SMR.1 – security roles Together the SFRs will result in protection of the management interfaces from unauthorized access.
O.MGMT_AUDIT	FAU_GEN.1 – audit events for the Management Console FAU_GEN.2 – user identity in audit events Together the SFRs will result in audited use of management interfaces.
O.KIOSK_AUDIT	FAU_GEN.1 – audit events for the Kiosk Scanning Functions FAU_GEN.2 – user identity in audit events Together the SFRs will result in audited use of scanning functions.
O.COMMS	FCS_COP.1 – cryptographic operations in support of communications FPT_ITT.1 – secure communication between the Kiosk and Core FTP_TRP.1 – secure communication with remote administrators

Objectives	SFR supports the objective by requiring:
	Together the SFRs will result in protected communications between TOE components and between the TOE and remote users.
O.UPDATE	FCS_COP.1 – cryptographic operations in support of trusted updated FPT_TUD.1 – digitally signed software updates Together the SFRs will result in authenticated software updates.

7.3 TOE Summary Specification Rationale

86 Table 21 provides a coverage mapping showing that all SFRs are mapped to the security functions described in the TSS.

Table 21: Map of SFRs to TSS Security Functions

	File Threat Analysis	Protected Communications	User Authentication	Security Management
FTR_SCN.1	X			
FTR_SCN.2	X			
FTR_CDR.1	X			
FTR_FVA.1	X			
FTR_DLP.1	X			
FTR_TAR.1	X			
FAU_GEN.1				X
FAU_GEN.2				X
FCS_COP.1		X		X
FIA_UAU.1			X	
FIA_UAU.7			X	
FIA_UID.1			X	

FMT_MOF.1				X
FMT_SMF.1	X			X
FMT_SMR.1				X
FPT_ITT.1		X		
FPT_TUD.1				X
FTP_TRP.1		X		