Draft Standard

MEF 162 Draft (R3)

Security Service Edge Certification Test Cases and Requirements

May 2024

This draft represents MEF work in progress and is subject to change.

Disclaimer

This draft document represents MEF work in progress; it has not achieved full MEF standardization and is subject to change. Changes are likely before this becomes a fully endorsed MEF Standard. The reader is strongly encouraged to keep this in mind and review the Release Notes (if applicable) when making a decision on adoption. Additionally, because this document has not been adopted as a Final Specification in accordance with MEF's Bylaws, Members are not obligated to license patent claims that are essential to implementation of this document under MEF's Bylaws.

Disclaimer

The information in this publication is freely available for reproduction and use by any recipient and is believed to be accurate as of its publication date. Such information is subject to change without notice and MEF Forum (MEF) is not responsible for any errors. MEF does not assume responsibility to update or correct any information in this publication. No representation or warranty, expressed or implied, is made by MEF concerning the completeness, accuracy, or applicability of any information contained herein and no liability of any kind shall be assumed by MEF as a result of reliance upon such information.

The information contained herein is intended to be used without modification by the recipient or user of this document. MEF is not responsible or liable for any modifications to this document made by any other party.

The receipt or any use of this document or its contents does not in any way create, by implication or otherwise:

a) any express or implied license or right to or under any patent, copyright, trademark or trade secret rights held or claimed by any MEF member which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
b) any warranty or representation that any MEF members will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
c) any form of relationship between any MEF member and the recipient or user of this document.

Implementation or use of specific MEF standards, specifications, or recommendations will be voluntary, and no Member shall be obliged to implement them by virtue of participation in MEF Forum. MEF is a non-profit international organization to enable the development and worldwide adoption of agile, assured and orchestrated network services. MEF does not, expressly or otherwise, endorse or promote any specific products or services.

Table of Contents

List of Figures

List of Tables

1 List of Contributing Members

The following members of the MEF participated in the development of this document and have requested to be included in this list.

Editor Note 1: *This list will be finalized before Letter Ballot. Any member that comments in at least one CfC is eligible to be included by opting in before the Letter Ballot is initiated. Note it is the MEF member that is listed here (typically a company or organization), not their individual representatives.*

- ABC Networks
- XYZ Communications

2 Abstract

Per Gartner, Security Service Edge (SSE) secures access to the web, cloud services, and private applications. Capabilities include access control, threat protection, data security, security monitoring, and acceptable-use control enforced by network-based and API-based integration. SSE is primarily delivered as a cloud-based service and may include on-premises or agent-based components.

The focus of this document is on the Test Cases and Requirements that are used to provide MEF Certification and Ratings. The certification of SSE implementation are seen as key for enterprise customers to have the knowledge about different offerings and solutions so that they can make informed decisions when purchasing SSE.

Note: MEF has no official SSE document but does describe many of the requirements for SSE in MEF 88 [8] and MEF 117 [10]. For the purposes of this document requirements are taken from those documents to be included in the SSE certification.

3 Release Notes

This document is currently out for Call for Comments Ballot number 4 and the contents of this document are subject to change based on comments received. All comments from Call for Comments Ballot 3 have been discussed and resolved.

4 Terminology and Abbreviations

This section defines the terms used in this document. In many cases, the normative definitions to terms are found in other documents. In these cases, the third column is used to provide the reference that is controlling, in other MEF or external documents.

In addition, terms defined in MEF 117 [10], MEF 118 [11], and MEF 138 r1 [12] are included in this document by reference and are not repeated in the table below.

Term	Definition	Reference
Access Control	A security technique that regulates who or what can view or use resources in a computing environment.	This document
Anti-Malware	A type of software program created to protect information technology (IT) systems and individual computers from malicious software, or malware.	This document
Authentication	The process or action of verifying the identity of a user or process.	This document
Cloud Access	A method that provides access to cloud-based resources	This document
Cloud Access/Application Control	Controls users' access to specific Cloud applications	This document
Compressors	A method of reducing data file size	This document
Data Loss Prevention	A security solution that identifies and helps prevent unsafe or inappropriate sharing, transfer, or use of sensitive data.	This document
Data Protection	A set of strategies and processes you can use to secure the privacy, availability, and integrity of your data.	This document
Decryption	The transformation of data that has been encrypted and rendered unreadable back to its unencrypted form	This document
Digital Experience Monitoring	The ability to view application performance issues from a user experience vantage point, isolate service performance problems across the delivery chain, speed up root cause determination and resolution and optimize digital transactions and customer journeys.	This document
DNS Protection	The concept of protecting the DNS service as a whole, sometimes with an emphasis on security.	This document
Encryption	The process of protecting information or data by using mathematical models to scramble it in such a way that only the parties who have the key to unscramble it can access it.	This document
Encrypted Traffic Inspection	The process of checking encrypted traffic by using the same technique as an on-path attack on the network connection	This document
Evasions	Attacks in which the attacker aims to manipulate the input data to produce an error in the machine learning system.	This document

Term	Definition	Reference
Exploit	A piece of software, data or sequence of commands that takes advantage of a vulnerability to cause unintended behavior or to gain unauthorized access to sensitive data.	This document
Exploit Detection and Prevention	Detects program code that takes advantage of vulnerabilities on the computer to exploit administrator privileges or to perform malicious activities.	This document
Firewall as a Service	A firewall solution delivered as a cloud-based service	This document
HTML Obfuscation	A means to encrypt or hide strings of characters which are likely email addresses and other source code within the code page.	This document
Identity	The fact of being the same person or thing as claimed	This document
Malicious Traffic	Traffic that represents incoming requests or a suspicious file's attempt to connect to an untrusted resource.	This document
Packers	Software that unpacks itself in memory when the "packed file" is executed.	This document
Policy Enforcement	The process of ensuring that the security policies and procedures implemented by an organization are followed consistently by its employees, partners, and stakeholders.	This document
Sandbox Protection	A cybersecurity practice where you run code, observe and analyze and code in a safe, isolated environment on a network that mimics end-user operating environments.	This document
Security Service Edge	Secures access to the web, cloud services and private applications.	This document
Threat Prevention	The ability to block specific threats before they penetrate the environment or before they do damage.	This document
Traffic Onboarding Methods		This document
Validation	Checking the accuracy and quality of source data before using, importing or otherwise processing data.	This document
Zero Trust	A security framework requiring all users, whether in or outside the organization's network, to be authenticated, authorized, and continuously validated for security configuration and posture before being granted or keeping access to applications and data.	This document

Table 1 – Terminology

Abbreviation	Definition	Reference
CASB	Cloud Access Security Broker	
CI/CD	Continuous Integration/Continuous Delivery	
DLP	Data Loss Prevention	

Abbreviation	Definition	Reference
FWaaS	Firewall as a Service	
IMAP	Internet Message Access Protocol	
IoT	Internet of Things	
MAPI	Messaging Application Programming Interface	
POP3	Post Office Protocol	
SFTP	Secure File Transfer Protocol	
SIEM/XDR	Security Information and Event Management/Extended Detection Response	
SMTP	Simple Mail Transfer Protocol	
SSE	Security Service Edge	
SSL/TLS	Secure Sockets Layer/Transport Layer Security	
SWG	Secure Web Gateway	

Table 2 – Abbreviations

5 Compliance Levels

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**NOT RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in BCP 14 (RFC 2119 [1], RFC 8174 [7]) when, and only when, they appear in all capitals, as shown here. All key words must be in bold text.

Items that are **REQUIRED** (contain the words **MUST** or **MUST NOT**) are labeled as **[Rx]** for required. Items that are **RECOMMENDED** (contain the words **SHOULD** or **SHOULD NOT**) are labeled as **[Dx]** for desirable. Items that are **OPTIONAL** (contain the words **MAY** or **OPTIONAL**) are labeled as **[Ox]** for optional**.**

Editor Note 3: The following paragraph will be deleted if no conditional requirements are used in the document.

A paragraph preceded by **[CRa]<** specifies a conditional mandatory requirement that **MUST** be followed if the condition(s) following the "<" have been met. For example, "**[CR1]<**[D38]" indicates that Conditional Mandatory Requirement 1 must be followed if Desirable Requirement 38 has been met. A paragraph preceded by **[CDb]<** specifies a Conditional Desirable Requirement that **SHOULD** be followed if the condition(s) following the "<" have been met. A paragraph preceded by **[COc]<** specifies a Conditional Optional Requirement that **MAY** be followed if the condition(s) following the "<" have been met.

6 Numerical Prefix Conventions

Editor Note 4: This section will be deleted if no numerical prefixes are used in the document.

This document uses the prefix notation to indicate multiplier values as shown in Table 3.

Decimal		**Binary**	
Symbol	**Value**	**Symbol**	**Value**
k	10^3	Ki	2^{10}
M	10^6	Mi	2^{20}
G	10^9	Gi	2^{30}
T	10^{12}	Ti	2^{40}
P	10^{15}	Pi	2^{50}
E	10^{18}	Ei	2^{60}
Z	10^{21}	Zi	2^{70}
Y	10^{24}	Yi	2^{80}

Table 3 – Numerical Prefix Conventions

7 Introduction

Per Gartner, Security Service Edge (SSE) secures access for people, places and things to the web, cloud services, and private applications. Capabilities include access control, threat protection, data security, security monitoring, acceptable-use control enforced by network-based and API-based integration and Protecting Users, workloads and devices. SSE is primarily delivered as a cloud-based service and may include on-premises or agent-based components.

The focus of this document is on the Test Cases and Requirements that are used to provide MEF Certification and Ratings. The certification of SSE implementations is seen as key for enterprise customers to have the knowledge about different offerings and solutions so that they can make informed decisions when purchasing SSE.

The certification testing defined within this document is intended to provide a rating, from D (lowest) to AAA (highest). Ratings are determined based on the results of the test cases defined in this document. Examples of these test cases would include the types of consumers (users, IoT devices etc.), inspection capability options (SWG, DLP, Anti-Malware etc.), and aspects of performance (digital experience monitoring).Ratings are based on weighting that is applied to each section of the test requirements in this document. See section 16 for more details on the rating methodology for SSE implementation.

After the completion of testing an overall rating is provided. This overall rating of an SSE implementation can be used to compare different SSE Vendor's ability to meet the test requirements, and therefore the requirements of Service Providers and enterprise customers. In the same manner, Service Provider's offerings can be compared to determine how different offerings address the end customer's requirements.

In addition to providing an overall rating, a MEF Certification is provided. This MEF Certification is based on compliance with MEF standards that are tested using the test methodologies defined in sections 14 and 15.

The testing defined within this document is intended to be repeatable to cover new software releases, service configurations, and updates to the way an SSE implementation is managed. The use of Continuous Integration/Continuous Deployment (CI/CD) strategies for MEF certification is being defined. Repeating the certification conformance and ratings process allows ratings to increase or decrease based on the performance of an implementation or service during continued testing. If a new software release breaks a critical function, this can be identified during repeated certification testing and the rating adjusted accordingly. In the same way, if a new software release provides fixes for shortfalls identified in previous certification testing, the rating can be increased accordingly.

7.1 Security Service Edge

The convergence of cloud computing, omnipresent and ubiquitous high-speed Internet, and tools that enable remote mobile workforces are changing how modern enterprises operate. This new paradigm is driving the need for scalable, flexible security architectures. SSE solutions leverage the cloud's scalability, flexibility, and operational benefits to deliver security functions such as –, Authentication and Identity, Data Loss Prevention (DLP), DNS Protocol Filtering (DPF), Encrypted Traffic Inspection (SSL/TLS), Exploit detection and prevention, Malware Detection and Removal (MD+R), and Cloud Access/Application control (CASB).

This document has categorized these SSE capabilities into the following:

- **Traffic onboarding methods**: client and clientless / proxy-based connectivity and edge connectivity options. Examples include TLS/SSL.
- **Firewall as a Service**: Identity-based Access Control, segmentation, decryption (TLS/SSL), ZTNA (per MEF 88 [8]).
- **Threat Prevention**: Exploit and Malware Prevention, Evasions
- **Data Protection**: Data Loss Prevention, Cloud Access / Application control (CASB)
- **Additional policy-based controls for applications [Recommended]**
- **External integrations (logging, SIEM / XDR support etc.) [Recommended]**

7.2 What will be tested?

The certification described in this document is designed to address the challenges faced by security and IT professionals in selecting and managing security products. The scope of the test methodologies in this document includes the following capabilities which are considered essential in any SSE offering:

- Test traffic onboarding methods
  - How traffic is onboarded
- Segmentation, Policy Enforcement and Access Control
  - How access is segmented, security policies are enforced, and how access to computing resources is controlled
- Decryption Validation and Bypass Exceptions
  - The decryption of traffic based on specific encryption methods and the ability to bypass decryption for certain traffic
- Threat Efficacy Testing
  - Testing to determine if threats can be detected
- Data Protection Validation
  - Ensuring that data can be encrypted, decrypted, and inspected
- Performance Impact, Redundancy & Monitoring Capabilities
  - The impact on the performance of a solution when the SSE implementation is stressed

Editor Note 5: Comments on the applicability of a test methodology to a specific SSE implementation are requested.

7.2.1 Test Configuration

The test configuration used to perform testing of SSE implementation is shown in Figure 1.

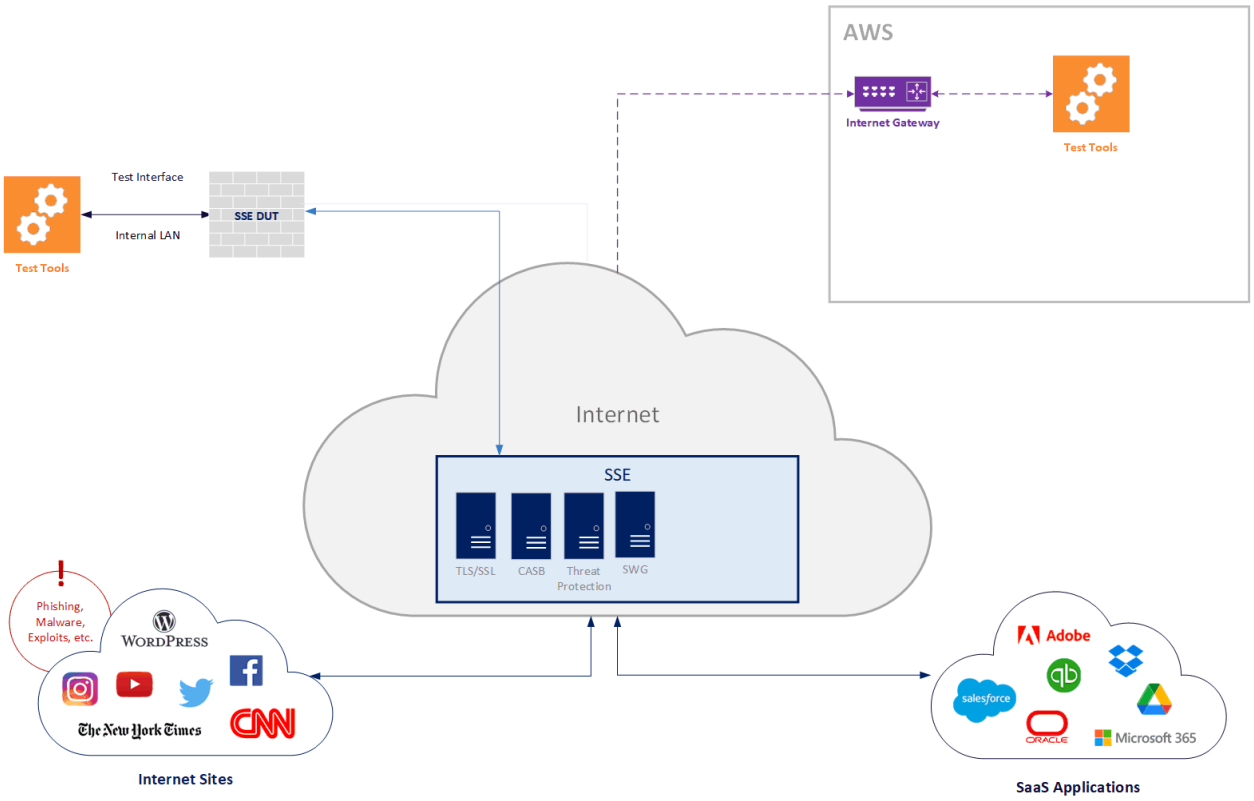

Figure 1 – Example Test Configuration for SSE

Figure 1 reflects the SSE functions within a single device that is providing security. The tests outlined in the remainder of this document use this basic configuration. The fact that security functions can be provided by different devices or cloud services is not reflected in the figure.

Note: There may be slight modifications to this test configuration for specific tests.

7.2.2 Testing Conventions

There are several conventions that are used for all test cases within this document. Every test case shows the Test Objective and Test Process to be used to perform the testing.

Where lists of protocols, evasions, applications, etc. are shown, this list is not intended to be a final list used for testing, instead, it provides samples of what will be tested. The test agreement contains the final list of what will be tested. This may include items that do not appear within this document.

Editor Note 6: The lists discussed above may be reduced in a future revision of this document so that Threat Actors cannot easily identify what is being tested or is not being tested. Your comments on this are requested.

8 Performance

The tests in this section measure the performance of a device using traffic scenarios that allow the inference of real-world performance. The SSE is the focus of this testing. Each test is performed concurrently without the SSE implementation to provide a baseline control. Results are reported both as measured, relative to the baseline, and in context with other measurable attributes and confounding variables.

There is no scoring penalty for any tests in section 8. These tests are performed to gather benchmarks for other tests in this document.

8.1 HTTP Traffic Analysis

The purpose of HTTP traffic analysis is to stress the detection engine in the SSE implementation to see how it copes with HTTP network loads of varying average packet size and varying connections per second. By creating session-based traffic with varying session lengths, the SSE implementation is forced to track valid TCP sessions, ensuring a higher workload than simple packet-based background traffic. The HTTP test traffic characteristics communicating directly over the Internet (i.e., no SSE implementation in the middle) are shown in Table 4.

Connections per Second per Gigabit	HTML Response Size in Bytes	Total Response Size in Bytes
1,000	115,570	129,738
2,000	57,388	64,824
4,000	28,048	32,136
8,000	13,512	15,920
16,000	6,353	7,916
32,000	2,667	3,903

Table 4 – HTTP Example Test Traffic Characteristics

8.1.1 HTTP Capacity (without transaction delays)

Test Objective: The purpose of these tests is to determine the performance delta for HTTP connections.

Test Process: Each transaction consists of a single HTTP GET request with no transaction delays (i.e., the web server responds immediately to all requests) with one TCP session being opened per session. All packets contain valid payloads. The tested Connections Per Second values are shown in Table 4.

[R1] The test **MUST** measure a benchmark for the number of HTTP connections per second per Gigabit that the SSE implementation supports.

Note: the measured benchmarks will be used in comparison to other measurements that are specified in this document.

8.1.2 HTTP Capacity (with transaction delays)

Test Objective: The purpose of these tests is to determine the performance delta for HTTP connections when introducing delays that more closely simulate actual activity. The ten second delay represents "looking time" for a web site by a user.

Test Process: Each transaction consists of a single HTTP GET request with delays (i.e., the web server responds immediately to all requests, but the client waits for 10 seconds before closing the connection). All packets contain valid payloads. The tested CPS values are shown in Table 4.

[R2] The test **MUST** measure the number of HTTP connections per second per Gigabit with transaction delays present as described above that the SSE implementation supports.

8.2 HTTPS Traffic Analysis

The purpose of these tests is to determine the performance curve and identify potential bottlenecks. The HTTPS detection engine is stressed to see how the device copes with network loads of varying average packet size and varying connections per second. By creating session-based traffic with varying session lengths, the device is forced to track valid TCP sessions, ensuring a higher workload for simple packet-based background traffic. This provides a test environment that simulates real-world behavior.

Protocol	Cipher Suite Description	Value	Frequency Ranking	Security Classification
TLS 1.3	TLS_AES_256_GCM_SHA384	(0x13, 0x02)	1	Recommended
TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	(0xC0, 0x30)	2	Secure

Table 5 – Cipher Suites for HTTPS Performance Tests

Table 5 shows the cipher suites that are used for testing of HTTPS. These are selected since they are currently the most used. In the future, the cipher suites are subject to change as these are replaced with more secure alternatives.

8.2.1 HTTPS Performance Delta TLS 1.3

Test Objective: The purpose of these tests is to determine the performance delta for HTTPS connections. The impact, if any, on performance when traffic is encrypted using the selected cipher suite (TLS_AES_256_GCM_SHA384 (0x13, 0x02)) is determined.

Test Process: Each transaction consists of a single HTTPS GET request with no transaction delays (i.e., the web server responds immediately to all requests). All traffic contains valid payloads. The HTTP test traffic characteristics using this cipher suite and communicating directly over the Internet (i.e., no SSE in the middle) are shown in Table 6.

Note: This test operates on the outer encryption. Support for TLS 1.3 is optional. This test is not performed if TLS 1.3 is not supported.

TLS_AES_256_GCM_SHA384 (0x13, 0x02)		
Connections per Second per Gigabit	HTML Response Size in Bytes	Total Response Size in Bytes
1,000	113,430	127,666
2,000	54,917	62,455
4,000	25,700	29,170
8,000	11,170	13,483
16,000	3,870	5,358
32,000	150	1,227

Table 6 – TLS 1.3 Cipher Suite Test Traffic Characteristics (0x13, 0x02)

[R3] The test **MUST** measure a benchmark for the number of HTTPS connections per second per Gigabit that the SSE, ZT, SASE implementation supports when using cipher suite TLS_AES_256_GCM_SHA384.

8.2.2 HTTPS Performance Delta TLS 1.2

Test Objective: The purpose of these tests is to determine the performance delta for HTTPS connections. The impact, if any, is determined when traffic is encrypted using the selected cipher suite (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xC0, 0x30)).

Test Process: Each transaction consists of a single HTTPS GET request with no transaction delays (i.e., the web server responds immediately to all requests). All traffic contains valid payloads. The HTTP test traffic characteristics using this cipher suite and communicating directly over the Internet (i.e., no SSE in the middle) are shown in Table 6.

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xC0, 0x30)		
Connections per Second per Gigabit	HTML Response Size in Bytes	Total Response Size in Bytes
1,000	115,000	129,360
2,000	56,257	63,945
4,000	26,970	31,047
8,000	12,394	14,808
16,000	5,047	6,738
32,000	1,365	2,605

Table 7 – TLS 1.2 Cipher Suite Test Traffic Characteristics (0xC0, 0x30)

Note: This test operates on the outer encryption. Support for TLS 1.2 is mandatory. This test is always performed.

[R4] The test **MUST** measure a benchmark for the number of HTTPS connections per second per Gigabit that the SSE implementation supports when using cipher suite TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.

8.3 Delta in Average Time to Transfer Data with and without SSE

Test Objective: Determine if the SSE under test has any impact on the time it takes to download files (e.g., PDFs, data files, zipped files, documents, etc.

Test Process: The same cipher selection methodology outlined in Section 8.2 will be used to determine testing targets under this section. The top two ciphers listed in Table 5 will be used. The same configuration established for testing under Section 8.2 of this methodology will be used to ensure that the device is not bypassing the decryption or inspection functionality within the SSE implementation. Tests using files in a variety of formats are performed with and without the SSE implementation offering to determine the impact of the test. Results are calculated relative to the baseline (i.e., the time difference between downloading files without SSE implementation and with the SSE implementation). To ensure a fair comparison, measurements are performed with and without the SSE implementation simultaneously, outliers are discarded, and the average time to download the files is calculated. The delta is calculated and recorded. See section 16 for details on any impact of the delta. File sizes and sources include the following sections.

Note: testing is performed with security functions enabled.

To eliminate any issues with performing these tests to the Internet, two paths, the Control path and the Path Under Test are used simultaneously to perform measurements.

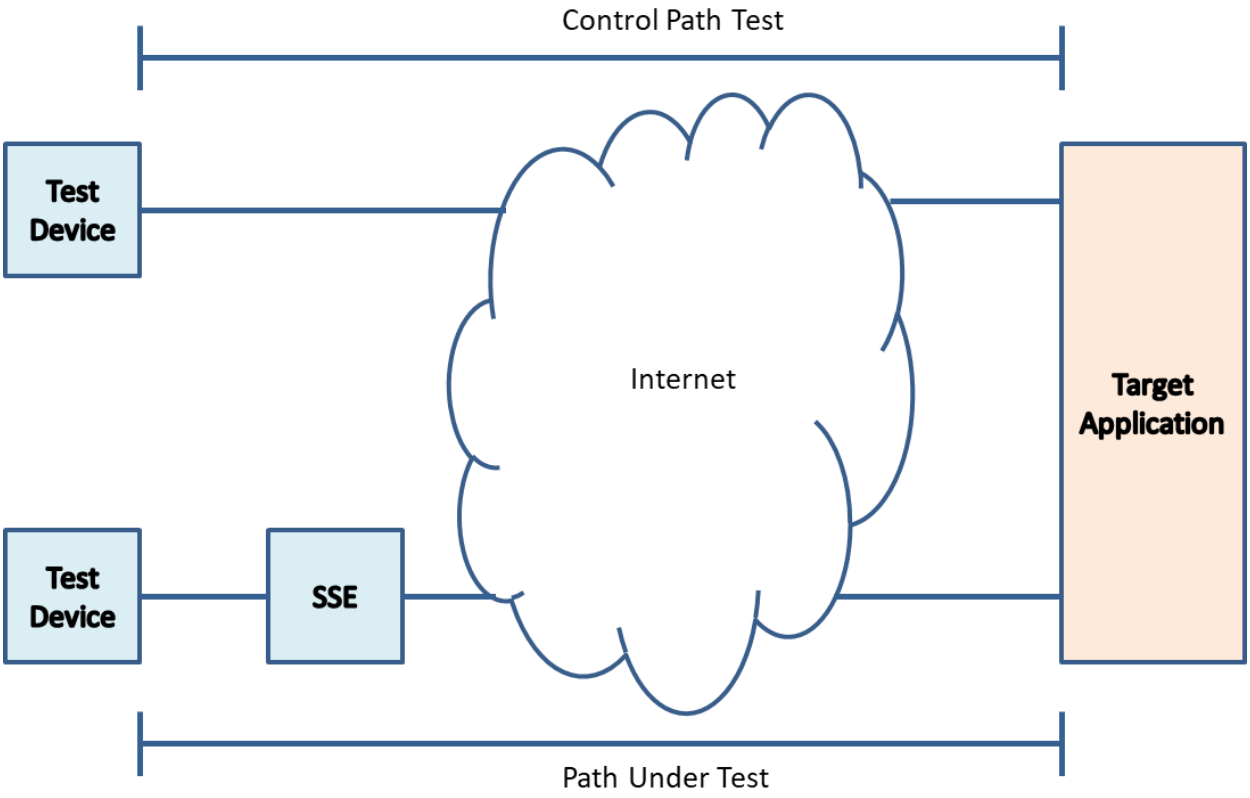

Figure 2 – Control Path and Path Under Test

Figure 2 shows that when tests of SSE file downloads are run to applications on the internet (Path Under Test), there is also a Control test (baseline) that is underway for the same application at the same time. The results of the tests on the Path Under Test are then compared to the results of the Control Path. The baseline is done without an SSE. Deltas between the measurements indicates the contribution of the SSE to the test device results. Outliers are removed from the results, and the target application. The path under test includes an SSE between the test device and the target application. The delta in performance between the Baseline and path under test is measured. Outliers [if any] are ignored.

Note: where encryption is used for the SSE framework, the SSE only decrypts and re-encrypts the outermost encryption layer.

8.3.1 File Size – 1MB

8.3.1.1 Microsoft OneDrive

8.3.1.2 Dropbox

8.3.1.3 Google Drive

8.3.1.4 HTTP web server

8.3.1.5 HTTPS web server (SSL/TLS)

8.3.2 File Size – 10MB

8.3.2.1 Microsoft OneDrive

8.3.2.2 Dropbox

8.3.2.3 Google Drive

8.3.2.4 HTTP web server

8.3.2.5 HTTPS web server (SSL/TLS)

8.3.3 File Size – 100MB

8.3.3.1 Microsoft OneDrive

8.3.3.2 Dropbox

8.3.3.3 Google Drive

8.3.3.4 HTTP web server

8.3.3.5 HTTPS web server (SSL/TLS)

8.3.4 File Size – 1,000MB

8.3.4.1 Microsoft OneDrive

8.3.4.2 Dropbox

8.3.4.3 Google Drive

8.3.4.4 HTTP web server

8.3.4.5 HTTPS web server (SSL/TLS)

9 SSL/TLS Support

To address the growing threat of focused attacks using the most common web protocols and applications, the capabilities of SSE to support a range of cipher suites and provide visibility into the encrypted payloads to detect attacks concealed by encryption as well as attacks against the encryption protocols themselves is verified. The cipher suites indicated within this section of the document are subject to change as the industry moves to new cipher suites.

9.1 Cipher Suite Support

To provide visibility into potential threats that are encrypted using SSL/TLS, the implementation being tested is expected to support a wide range of commonly used cipher suites. Cipher suites are selected based on the published current frequency of use[1] and security status[2]. The cipher suite used in testing will be specified as a part of the test agreement. The top 10 ciphers are expected to be included in the testing.

[1] Published international daily cipher suite usage can be found at https://crawler.ninja/files/ciphers.txt

[2] A list of cipher suites and associated attributes including security ratings can be found at https://ciphersuite.info/cs/

Protocol	Cipher Suite Description	Value	Frequency Ranking	Security Classification
TLS 1.3	TLS_AES_256_GCM_SHA384	(0x13, 0x02)	1	Recommended
TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	(0xC0, 0x30)	2	Secure
TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA256	(0xC0, 0x2F)	3	Secure
TLS 1.3	TLS_AES_128_GCM_SHA256	(0x13, 0x01)	4	Recommended
TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	(0xC0, 0x28)	5	Weak
TLS 1.2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256	(0xCC, 0xA8)	6	Secure
TLS 1.3	TLS__CHACHA20_POLY1305_SHA256	(0x13, 0x03)	7	Recommended
TLS 1.2	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	(0xC0, 0x2C)	8	Recommended
TLS 1.2	TLS_ECDHE_EDCSA_WITH_CHACHA20_POLY1305_SHA256	(0xCC, 0xA9)	9	Recommended
TLS 1.2	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384	(0x00, 0x9F)	10	Secure
TLS 1.2	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	(0xC0, 0x2B)	11	Recommended
TLS 1.0	TLS_DHE_RSA_WITH_AES_256_CBC_SHA	(0x00, 0x39)	12	Weak
TLS 1.2	TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	(0x00, 0xA2)	N/A	Recommended
TLS 1.2	TLS_ECDHE_EDCSA_WITH_ARIA_128_GCM_SHA256	(0xC0, 0x5C)	N/A	Recommended
TLS 1.2	TLS_DHE_DSS_WITH_ARIA_128_GCM_SHA256	(0xC0, 0x56)	N/A	Recommended
TLS 1.2	TLS_ECDHE_EDCSA_WITH_ARIA_256_GCM_SHA384	(0xC0, 0x5D)	N/A	Recommended
TLS 1.2	TLS_DHE_DSS_WITH_ARIA_256_GCM_SHA384	(0xC0, 0x57)	N/A	Recommended
TLS 1.2	TLS_DHE_DSS_WITH_AES_256_GCM_SHA384	(0x00, 0xA3)	N/A	Recommended

Table 8 – Cipher Suites

9.1.1 Current Cipher Suites

Based on the different cipher suites that exist, the SSE implementation support for current cipher suites is determined using this test. The purpose of this test is to determine which cipher suites are supported. Tested cipher suites are selected based on frequency of use and security recommendations from reputable sources. The cipher suites available for this test include those listed in Table 8. Since this test is minimally invasive, it will be randomly repeated during the test period with multiple SSE implementation instances to confirm standard availability.

[R5] The test **MUST** verify which cipher suites described in Table 8 are supported by the SSE implementation.

Note: If a cipher suite is not supported, there must be a method to block that cipher suite.

9.1.2 Insecure Cipher Suites

Some cipher suites do not provide secure encryption. These are indicated as WEAK in Table 8. The purpose of the testing is to determine how the SSE implementation handles cipher suites known to be insecure including the following:

- Null cipher suites (no Encrypted Traffic Inspection of data provided)
- Anonymous cipher suites (no key authentication provided)

The insecure cipher suites are expected to be blocked.

[R6] The test **MUST** verify how the SSE implementation handles cipher suites are considered by this document to be insecure.

9.1.3 Scoring Penalty

The scoring penalty for sections 9.1.1 and 9.1.2 is 100%.

9.2 Encrypted Stream Validation

The ability of an SSE implementation to correctly decrypt and inspect SSL/TLS traffic prior to the associated performance testing described in this document is verified by this testing. The SSE implementation is expected to support all test cases with a single configuration. The purpose of this test is to use content which has been successfully blocked by the SSE implementation during testing. The content is then embedded in encrypted traffic to determine the capabilities of the SSE implementation to enforce policy for encrypted streams.

[R7] This test **MUST** verify the ability of the SSE implementation to determine if the implementation can enforce Policy for encrypted streams.

9.2.1 Scoring Penalty

The scoring penalty for section 9.2 is 100%.

9.3 Decryption Bypass Exceptions

The SSE implementation is expected to support the configuration of policies that permit conditional bypass of decryption to preserve privacy, either for regulatory or other reasons (examples could be related to banking or medical data). The purpose of this test is to verify that the SSE implementation maintains decryption capabilities concurrently with inspection exception rules. The test will verify that specific traffic, ciphers, or IP Addresses are excluded from decryption. (Turning off all decryption on the SSE implementation would not be an acceptable method for meeting this requirement.)

[R8] This test **MUST** verify the ability of the SSE implementation to perform decryption concurrently with inspection exception rules.

9.3.1 Scoring Penalty

The scoring penalty for section 9.3 is 100%.

10 Management Capabilities

It is important that an SSE implementation provide comprehensive management control to accomplish the expected functionality. Further best practices in user experience should be fundamental to the associated interface.

10.1 Authentication

Editor Note 7: Additional options will be added to this section after the completion of the Beta testing.

10.1.1 Role-Based Access Control

Test Objective: This test verifies that RBAC is supported by the SSE. Role Based Access Control (RBAC) is defined in MEF 118 as "A collection of access Authorizations a Subject or Target Entity receives based on a given set of Roles."

Test Process: This test is performed by creating Roles for Subject and Target Entities and ensuring that authentication is done correctly.

[R9] The test **MUST** verify that RBAC is supported for Subject and Target Entities.

10.1.1.1 Scoring Penalty

The scoring penalty for section 10.1.1 is 100%.

10.2 Policy

10.2.1 Policy Definition

Test Objective: This test verifies that the SSE implementation allows the creation of policies that are used to control access, functionality, and behavior of an SSE implementation.

Test Process: The policies will be tested to encrypt all, decrypt all, and pass selected without decrypting. This is performed by creating policies within the SSE implementation that allows the Encrypt all, Decrypt all and do not Decrypt specific protocols.

Note: The protocols shown in this section are shown as examples. The specific list of protocols included in the testing will be defined in the Testing Agreement.

10.2.1.1 Protocols

- Encrypt, and Decrypt (Encrypt all, Decrypt all, do not Decrypt individual protocols) basic Internet services using at least 3 of the following protocols
    - HTTPS
    - SMTP
    - IMAP
    - POP3

- Exchange/MAPI
- SFTP

10.2.1.1.1 Scoring Penalty

The scoring penalty for section 10.2.1.1 is 100% for the applications defined in the Test Agreement.

10.2.1.2 Applications

- Encrypt and Decrypt (Encrypt all, Decrypt all, Decrypt Bypass) Subject Entities (internal and remote) connectivity via proxied connections to trusted 3rd party Target Entities using at least 3 of the following applications:
  - Office 365
  - Salesforce
  - NetSuite
  - Google Workspace
    - Google Drive (used for Decrypt Bypass)
  - Dropbox

Some applications such as Facebook Messenger and Facebook Application only support pining and are only tested for Decrypt Bypass.

10.2.1.2.1 Scoring Penalty

The scoring penalty for section 10.2.1.2 is 100% for the applications defined in the Test Agreement.

10.2.1.3 Locations

- Encrypt and Decrypt (Encrypt all, Decrypt all, do not Decrypt individual locations) Subject Entities at one location access to the tested Target Entities at another location
  - Enterprise Site to/from SSE
    - Outer Envelope
      - Encrypt all
      - Decrypt all
    - Payload
      - Encrypt all

- Decrypt all
- Decrypt bypass

- SSE to/from Cloud resources
  - Outer Envelope
    - Encrypt all
    - Decrypt all
    - No encryption
  - Payload
    - Encrypt all
    - Decrypt all
    - Decrypt bypass
-

Editor Note 8: Cloud-to-cloud will be added here after the Beta is completed.

10.2.1.3.1 Scoring Penalty

The scoring penalty for section 10.2.1.3 is 100% for the applications defined in the Test Agreement.

10.2.1.4 Social Media

- Encrypt and Decrypt (Encrypt all, Decrypt all, Decrypt Bypass individual social media applications and websites) popular social networking applications and websites
  - Top 10 social media applications including Facebook, Twitter, LinkedIn, Glassdoor, or other Web applications

Note: The top 10 social media applications will be updated every six months. This list will be updated, with social media applications added or deleted from the list as appropriate.

Note: that some Social Media applications only support pining which means that they are tested for Decrypt Bypass only. If pining is not supported, or optionally supported, Encrypt all and Decrypt all will also be tested.

10.2.1.4.1 Scoring Penalty

The scoring penalty for section 10.2.1.4 is 5% for the applications defined in the Test Agreement.

10.2.1.5 Video and Voice

- Encrypt and Decrypt (Encrypt all, Decrypt all, do not Decrypt individual teleconferencing applications) Video and Voice teleconferencing at least 3 of the following applications are tested
    - Microsoft Teams
    - Zoom
    - Cisco WebEx
    - Google Meet

10.2.1.5.1 Scoring Penalty

The scoring penalty for section 10.2.1.5 is 5% for the applications defined in the Test Agreement.

10.2.1.6 Applications and Websites

- Encrypt, and Decrypt (Encrypt all, Decrypt all, do not Decrypt individual applications and websites) streaming media applications and websites at least 3 of the following applications and websites are tested
    - Netflix
    - Prime Video
    - Hulu
    - YouTube
    - TikTok
    - Max/HBO Max
    - Disney+
    - AppleTV

[R10] The test **MUST** verify that policies that describe the behavior of SSE implementations can be created.

10.2.1.6.1 Scoring Penalty

The scoring penalty for section 10.2.1.6 is 5% for the applications defined in the Test Agreement.

10.2.2 Policy Validation

Testing in this section validates that policy has passed or blocked traffic as appropriate.

10.2.2.1 Protocols

Test Objective: Determine if the SSE implementation provides the ability to identify whether a policy has blocked or passed a specific protocol.

Test Process: The selected protocols are introduced at an SSE and the test verifies that they are passed to the appropriate end device or blocked by the policy.

[R11] The test **MUST** verify that when an IP Packet is allowed, the IP Packet passes to the destination.

[R12] The test **MUST** verify that when an IP Packet is blocked, the IP Packet does not reach the destination.

10.2.2.1.1 Scoring Penalty

The scoring penalty for section 10.2.2.1 is 5%.

10.2.2.2 Applications

Test Objective: Determine if the SSE implementation provides the ability to identify whether a policy has blocked or passed a specific application.

Test Process: The selected applications are introduced at an SSE and the test verifies that they are passed to the appropriate end device or blocked by the policy.

[R13] The test **MUST** verify that when an IP Packet is allowed, the IP Packet passes to the destination.

[R14] The test **MUST** verify that when an IP Packet is blocked, the IP Packet does not reach the destination Scoring Penalty

10.2.2.2.1 Scoring Penalty

The scoring penalty for section 10.2.2.2 is 5%.

10.2.2.2.2 Social Media

Test Objective: Determine if the SSE implementation provides the ability to identify whether a policy has blocked or passed a specific social media application.

Test Process: The selected social media applications are introduced at an SSE and the test verifies that they are passed to the appropriate end device or blocked by the policy.

[R15] The test **MUST** verify that when an IP Packet is allowed, the IP Packet passes to the destination.

[R16] The test **MUST** verify that when an IP Packet is blocked, the IP Packet does not reach the destination.

10.2.2.2.2.1 Scoring Penalty

The scoring penalty for section 10.2.2.2.2 is 5%.

10.2.2.2.3 Video and Voice

Test Objective: Determine if the SSE implementation provides the ability to identify whether a policy has blocked or passed a video or voice.

Test Process: The video and voice traffic are introduced at an SSE and the test verifies that they are passed to the appropriate end device or blocked by the policy.

[R17] The test **MUST** verify that when an IP Packet is allowed, the IP Packet passes to the destination.

[R18] The test **MUST** verify that when an IP Packet is blocked, the IP Packet does not reach the destination.

10.2.2.2.3.1 Scoring Penalty

The scoring penalty for section 10.2.2.2.3 is 5%.

10.2.2.2.4 Applications and Web Sites

Test Objective: Determine if the SSE implementation provides the ability to identify whether a policy has blocked or passed a specific application or web site.

Test Process: The selected applications or web sites are introduced at an SSE and the test verifies that they are passed to the appropriate end device or blocked by the policy.

[R19] The test **MUST** verify that when an IP Packet is allowed, the IP Packet passes to the destination.

[R20] The test **MUST** verify that when an IP Packet is blocked, the IP Packet does not reach the destination Scoring Penalty

10.2.2.2.4.1 Scoring Penalty

The scoring penalty for section 10.2.2.2.4 is 5%.

10.2.2.3 Locations

Test Objective: Determine if the SSE implementation provides the ability to identify whether a policy has blocked or passed a specific location.

Test Process: The selected locations pass traffic, and the test verifies that they are passed to the appropriate SD-WAN UNI or blocked by the policy.

[R21] The test **MUST** verify that when an IP Packet is allowed, the IP Packet passes to the destination.

[R22] The test **MUST** verify that when an IP Packet is blocked, the IP Packet does not reach the destination Scoring Penalty

10.2.2.3.1 Scoring Penalty

The scoring penalty for section 10.2.2.3 is 5%.

10.2.3 Policy Versioning

Test Objective: Determine of the SSE implementation provides the ability to secure Policies through the use of versioning.

Test Process: A policy is created to Decrypt all applications as shown above. This policy is then saved. Once saved, the policy is updated to not decrypt Google Workspace. Updating of this policy must require the appropriate permissions to make this update.

[R23] The test **MUST** verify that the SSE implementation manages versioning of Policies.

10.2.3.1 Scoring Penalty

The scoring penalty for section 10.2.3 is 25%.

11 Reporting Capabilities

Logging, alerting, and reporting are critical functions that inform the security posture and facilitate incident response actions. Reporting capabilities will be assessed to determine the ability of the SSE implementation to support these requirements.

11.1 Logs

Test Objective: Vereify that the SSE implementation allows generation and exportation of reports, logs, and alerts into industry-standard formats in support of incident response. (Aspects like log time normalization, and log file maintenance options will be factored in the assessment.)

Test Process: The events shown below are caused to occur and the SSE capability to generate the appropriate logs is determined.

[R24] The test **MUST** verify that Logs contain the following information:

- Date of occurrence
- Device event occurred on
- Subcomponent involved
- Detailed message

The SSE implementation collects and stores information about the following events:

- Malicious Traffic
- Administrator Login/Logout
- Successful Authentication
- Unsuccessful Authentication
- Policy Changes
- Policy Deployment
- Disk quota is close to being exceeded
- CPU utilization exceeds a specific threshold

The SSE implementation is checked to ensure that it creates a log for each of these events and that the log includes a date/time stamp.

[R25] The test **MUST** verify that a SSE implementation collects and stores information about the events shown in section 11.1.

11.1.1 Scoring Penalty

The scoring penalty for section 11.1 is 50%.

11.2 Security Event Notification

This section addresses Security Event Notifications

11.2.1 Security Event Notification Generation

Test Objective: The objective of this test is to determine if the SSE implementation generates Security Event Notifications (SENs) when specified events occur. SENs are to be provided, at a minimum, when malicious traffic is identified and blocked or when authentications are unsuccessful.

Test Process: Known malicious traffic that the SSE implementation has previously blocked are sent and it is ensured that a SEN is generated.

[R26] The test **MUST** verify that an SSE implementation generates SENs as described in section 11.2.1.

Editor Note 9: Additional SENs that cover other aspects of system availability and performance will be added after the beta testing is completed.

11.2.1.1 Scoring Penalty

The scoring penalty for section 11.2.1 is 100%.

11.2.2 SEN Filtering

Test Objective: The objective of this test is to determine if SSE implementations support filtering of SEN contents and summaries for selected attributes (e.g., view all SENs for a selected source IP).

Test Process: SENs are filtered on malicious traffic received and the receipt of malicious traffic performed previously is verified to be included in the SEN.

[R27] The test **MUST** verify that a SSE implementation allows SEN filtering.

11.2.2.1 Scoring Penalty

The scoring penalty for section 11.2.2 is 10%.

11.2.3 View SEN Detail

Test Objective: This test verifies if SSE implementations provide capabilities for in-depth information about SENs.

Test Process: The detail includes information about a specific SEN. The filtered SEN is reviewed to determine additional information beyond the date/time stamp that is included.

[R28] The test **MUST** verify that a SSE implementation allows the retrieval of SEN detail.

11.2.3.1 Scoring Penalty

The scoring penalty for section 11.2.3 is 10%.

11.2.4 SEN Suppression

Test Objective: This test verifies that SSE implementations provide the functionality to create exception filters based on SEN data to eliminate further SENs that match the specified criteria (e.g., same SEN ID from the same source IP).

Note: This does not disable detection, logging, or blocking but merely excludes SENs from the console display.

Test Process: A policy is changed which creates a log file entry but does not generate an SEN.

[R29] The test **MUST** verify that a SSE implementation allow SENs to be suppressed from the console display without impacting the detection, logging, or blocking of threats.

11.2.4.1 Scoring Penalty

The scoring penalty for section 11.2.4 is 5%.

11.3 Change Control

It is essential for the system to track, retain, and report changes to policies and rules.

Test Objective: The test verifies if Subject and Target Entities are monitored and logged, and, if possible, change management controls are implemented. These items fall under compliance process controls for change management, onboard and off-board, segregation of duties, and access control.

Test Process: Policies are created to Encrypt all applications as shown above. That policy is saved. The policy is then changed to also Decrypt all applications. The Change Control log is checked to verify an update. The ability to Roll-Back to the previous revision of the policy is verified. The Revision History is checked to ensure that each revision is included in the history.

Change Control functionality and capabilities include support for each of the following:

- Change Control Logging
- Roll-Back
- Revision History

[R30] The test **MUST** verify that change control is supported by the SSE implementation.

11.3.1 Scoring Penalty

The scoring penalty for section 11.3 is 15%.

12 Threat Prevention

Threat prevention tests assess how accurately the SSE implementation blocks and logs threats while remaining resistant to false positives. To accomplish this goal, the SSE implementation will be deployed using vendor-recommended settings. Protection being tested must be available to Subscribers at the time of testing. If Browser Isolation is available, vendors may choose to have their SSE implementation tested both with and without Browser Isolation. Threat prevention is tested with the SSE acting as a proxy.

12.1 False Positives

False positives are any legitimate, non-malicious traffic that the SSE implementation perceives as malicious and blocks. The ability to correctly identify and allow legitimate traffic while maintaining protection against attacks is a key to effective protection. False positive tests examine the ability of the SSE implementation to block attacks while permitting legitimate traffic. Implementations that block legitimate traffic will have the sensitivity of those protections turned down or disabled in order to allow legitimate traffic. Testing will determine which protections (e.g., signatures) trigger false positives. Signatures that trigger false positives must be disabled before the security testing begins. Consequently, the false positive test is repeated until no false positive signatures are detected, and all legitimate traffic passes.

12.1.1 Initial check – legitimate traffic, documents, and files

Test Objective: This test verifies if the SSE implementaton can decrypt and pass specific file formats.

Test Process: This test transmits a varied sample of legitimate application traffic, documents, and files which should be identified and allowed verifying that the SSE implementation can decrypt and recognize these file types. Testing includes some of the following file formats:

- HTML
- .exe
- .jar
- .xlsm
- .css
- .pdf
- .ppt
- .pptx
- .doc
- .docx
- .zip

- 7zip
- Gzip
- .DLL
- .js
- .xls
- .xlsx
- .chm
- .rar
- .Ink
- .cur
- .tar
- .xrc

[R31] The test **MUST** verify that legitimate traffic as shown in section 12.1.1 are passed.

12.1.1.1 Scoring Penalty

The scoring penalty for section 12.1.1 is 100%.

12.1.2 Ongoing check – legitimate traffic, documents, and files

Test Objective: The objective of this test is to generate legitimate traffic and verify that it can be decrypted and passed as described in sections 12.2, 12.3, 12.4, and 13.

Test Process: Testing will introduce legitimate traffic, documents, and files into in sections 12.2, 12.3, 12.4, and 13. Testing includes the following file formats:

- HTML
- .exe
- .jar
- .xlsm
- .css
- .pdf

- .ppt
- .pptx
- .doc
- .docx
- .zip
- 7zip
- Gzip
- .DLL
- .js
- .xls
- .xlsx
- .chm
- .rar
- .Ink
- .cur
- .tar
- .xrc

[R32] The testing performed in later sections of this document **MUST** include legitimate traffic as shown in section 12.1.2.

12.1.2.1 Scoring Penalty

The scoring penalty for section 12.1.2 is 100%.

12.2 Exploits

An exploit is an attack that takes advantage of a vulnerability in a protocol, product, operating system, or application.

Test Objective: Testing verifies that the SSE implementation is capable of detecting and blocking exploits while remaining resistant to false positives by attempting to send exploits through the product under test; and verifying that the malicious traffic is blocked, and all appropriate logging and notifications are performed.

Test Process: The exploit repository used for testing contains thousands of exploits over a wide range of protocols and applications. Exploit sets for individual tests are selected based on Common Vulnerability Scoring System (CVSS) score, use case. and relevance to customers.[3] Testing will determine if exploit traffic is blocked by the SSE implementation, and if the event is recorded in the log. All tests are performed with varying levels and mixes of background network load.

The exploits that are currently used for testing will be communicated to the SSE implementation Vendor before testing is performed. This list is subject to change as new exploits detected.

[R33] The test **MUST** verify that exploits are blocked while legitimate traffic is passed.

12.2.1 Scoring Penalty

The scoring penalty for section 12.2 is 100%.

12.3 Malware (Wild) Delivered over the HTTP/HTTPS

Users may be deceived into clicking on a malicious link (on, for example, a web page or a banner advertisement) to download and execute malware. In cases where an attacker is aiming for a large number of victims, the attacker may hijack widely used reputable websites to distribute the malware. This malware may be passed via clear text and SSL/TLS encrypted traffic while also passing legitimate (clear text and SSL/TLS encrypted) traffic.

Test Objective: Testing will determine if malware traffic is blocked by the SSE implementation, and if the event is recorded in the log. All tests are performed with varying levels and mixes of background network load. A test environment to assess the protection capabilities of SSE implementations under the most real-world conditions possible has been created. Testing uses a proprietary live testing harness that is scalable and capable of running thousands of concurrent endpoints.

[3] Vendors will be provided with a baseline set of malicious traffic prior to testing. These baseline samples will be used to verify basic protection capabilities and will not be part of the actual test.

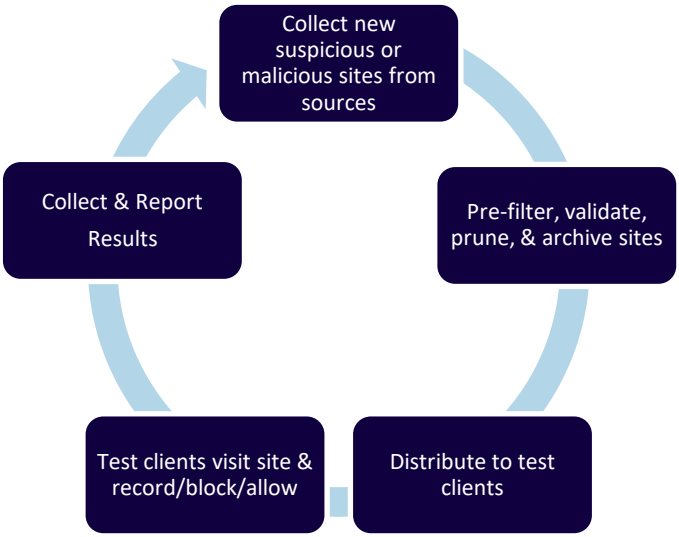

Figure 3 – Malware Test Workflow

Test Process: Figure 3 reflects the steps in this Malware testing. It starts at the top of the figure with collecting new or suspicious or malicious sites from sources. These are then included in test traffic. The test traffic is distributed to clients who are being certified. The client is able to perform testing using this malware and the results are collected and reported. These steps repeat as new malware is detected.

[R34] The test **MUST** verify that Malware is blocked while legitimate traffic is passed.

[R35] Testing of new Malware **MUST** be tested by SSE implementations using traffic obtained as described in sections 12.3.2, 12.3.5, 12.3.3, and 12.3.4.

Editor Note 10: The testing for Malware uses the sample methods described in the following sections. The scoring for this testing is for all types of Malware. If any are not blocked the testing is considered failed. Comments on this are requested.

12.3.1 Scoring Penalty

The scoring penalty for section 12.3 is 10% per missing exploit.

Editor Note 11: This has been changed from a 100% failure if any exploit fails to a penalty per exploit that fails. Comments are requested on this.

12.3.2 URL / Malware Sample Sets

To keep the Malware sample sets up to date, the test house maintain relationships with other independent security researchers, professional networks, and security companies from which they harvest live malware URLs. Throughout the test, new URLs are added as they are discovered. Sample sets contain malware URLs distributed via email, messaging, social networks, and websites. In addition, the test house operates a network of spam traps and honeypots yielding unique samples.

12.3.3 URL Status Confirmation

Given the nature of the feeds and the rate of change, it is not possible to validate each site in depth before the test, since many sites quickly disappear. However, each URL receives an initial review to verify that it meets basic test criteria and is accessible on the Internet at the time of testing. To be included in the test set, URLs must be live during each iteration of the test. At the beginning of each iteration, the availability of the URL is confirmed by ensuring that the site can be reached and is active (for example, a non-404 web page is returned). The active URL content is downloaded and saved to an archive server with a unique test ID number. This enables the test house to preserve the URL content for control and validation purposes. Note: every sample is validated after the test, and URLs are reclassified and/or removed accordingly.

12.3.4 Pruning and Validation

Throughout the test, the test house engineers review and remove non-conforming URLs and content from the test set. For example, a URL that initially was classified as phishing but that has since been replaced with a generic splash page will be removed and will not be included in future calculations. The test house continually verifies that each phishing and malware site is accessible and serving malicious content. Sites that are not available are not included in calculations of success or failure; however, if they become available during the test, those iterations will be included in calculations. Post-test validation enables the test house to reclassify and even remove phishing / malware sites that are not malicious or that were not available during the test.

12.3.5 URL Cataloging

All URLs under consideration are cataloged with a unique ID, regardless of their validity. Prompt and accurate URL cataloging enables the test house engineers to monitor the quality of sample sources and simplifies investigation and analysis. New sites are added to the URL test set as soon as possible following initial discovery. The date and time that each URL is introduced is recorded. Most sources are automatically and immediately inserted, while some methods require manual handling and can be processed in a timely manner. URLs that are either no longer reachable or hosting malware are removed from the test (but are maintained in the URL catalog).

12.4 Handcrafted Malware (Sandbox Protection)

The goal of this test is to determine which SSE implementations are able to protect customers while under adverse conditions dictated by the attacker. For the purposes of this test, handcrafted (targeted) malware is created by modifying the source code of keyloggers, ransomware, destructoware, etc. We then recompile the binary so that it is new to the SSE being tested. The test then attempts to infect a host (e.g., a laptop) with the malware and recorded whether or not the SSE implementation blocked the attack. Because creating samples in this manner is a painstaking and time-consuming exercise, only 1010 targeted samples will be tested; results should be viewed with this in mind.

12.4.1 Scoring Penalty

The scoring penalty for section 12.4 is 50%.

13 Evasions

Attackers use evasion techniques to disguise and modify attacks at the point of delivery to avoid detection by security products**.** Passing evasion tests is very important since just one successful evasion technique can enable an attacker to exploit systems undetected. Previously blocked exploits will be rerun using evasion techniques.

Testing verifies that the SSE implementation is capable of detecting and blocking exploits and malware when subjected to varying common evasion techniques. Wherever possible, the implementation is expected to successfully decode the obfuscated traffic to provide an accurate alert relating to the original attack, rather than alerting purely on anomalous traffic detected as a result of the evasion technique itself. There are no passwords included in the IP Packets. SSE solution vendors are expected to detect evasions as specified in sections 13.1, 13.2, 13.3, 13.4, and 13.5.

Editor Note 12: The below evasions are updated as they are detected. The list of current evasions is provided to the SSE implementation Vendor prior to testing.

13.1 HTTP Obfuscation

Web browsers request content from servers over HTTP using the ASCII character-set. HTTP encoding replaces unsafe non-ASCII characters with a "%" followed by two hexadecimal digits. Web servers and clients understand how to decode the request and responses. However, this mechanism can be abused to circumvent protection that is looking to match specific strings of characters.

Chunked encoding allows the server to break a document into smaller chunks and transmit them individually. The server needs only to specify the size of each chunk before it is transmitted and then indicate when the last chunk has been transmitted. Since chunked encoding intersperses arbitrary numbers (chunk sizes) with the elements of the original document, it can be used to greatly change the appearance of the content as observed "on the wire" during transmission. In addition, the server can choose to break the document into chunks at arbitrary points. This makes it difficult to reliably identify the original HTML content from the raw data on the network.

Test Objective: This test is to verify that evasions within HTTP can be detected.

Test Process: Evasions including manipulation using the the following techniques will be tested:

- Declared HTTP/0.9 response; but includes response headers; chunking declared but served without chunking
- HTTP/1.1 chunked response with chunk sizes preceded by multiple zeros (hex '30')
- HTTP/1.1 chunked response with chunk sizes followed by backspace (hex '08')
- HTTP/1.1 chunked response with chunk sizes followed by end of text (hex '03')
- HTTP/1.1 chunked response with chunk sizes followed by escape (hex' 1b')
- HTTP/1.1 chunked response with chunk sizes followed by null (hex '00')

- HTTP/1.1 chunked response with chunk sizes followed by a space (hex '20') then a zero (hex '30')
- HTTP/1.1 chunked response with final chunk size of '00000000000000000000000000000000000000000000000000000000000000000000000000000000' (rather than '0')
- HTTP/1.1 response with line folded transfer-encoding header declaring chunking ('Transfer-Encoding: ' followed by CRLF (hex '0d 0a') followed by 'chunked' followed by CRLF (hex '0d 0a'); served without chunking
- HTTP/1.1 response with transfer-encoding header declaring chunking with lots of whitespace ('Transfer-Encoding:' followed by 8000 spaces (hex '20' * 8000) followed by 'chunked' followed by CRLF (hex '0d 0a'); served chunked
- HTTP/1.0 response declaring chunking; served without chunking
- HTTP/1.0 response declaring chunking with invalid content-length header; served without chunking
- HTTP/1.1 response with "\tTransfer-Encoding: chunked"; served chunked
- HTTP/1.1 response with "\tTransfer-Encoding: chonked" after custom header line with "chunked" as value; served without chunking
- HTTP/1.1 response with header with no field name and colon+junk string; followed by '\tTransfer-Encoding: chunked' header; followed by custom header; served chunked
- HTTP/1.1 response with "\r\rTransfer-Encoding: chunked"; served chunked
- HTTP/1.1 response with using single "\n"'s instead of "\r\n"'s; chunked
- HTTP/1.1 response with \r\n\r\n before first header; chunked
- HTTP/1.1 response with "SIP/2.0 200 OK\r\n" before status header; chunked
- HTTP/1.1 response with space+junk string followed by \r\n before first header; chunked
- HTTP/1.1 response with junk string before status header; chunked
- HTTP/1.1 response with header end \n\014\n\n; chunked
- HTTP/1.1 response with header end \r\n\016\r\n\r\n; chunked
- HTTP/1.1 response with header end \n\r\r\n; chunked
- HTTP/1.1 response with header end \n\017\018\n\n; chunked
- HTTP/1.1 response with header end \n\030\n\019\n\n; chunked
- HTTP/1.1 response with status code -203.030; with message-body; chunked

- HTTP/1.1 response with status code 402; with message-body; chunked
- HTTP/1.1 response with status code 403; with message-body; chunked
- HTTP/1.1 response with status code 406; with message-body; chunked
- HTTP/1.1 response with status code 505; with message-body; chunked
- HTTP/1.1 chunked response with no status indicated
- No status line; chunking indicated; served unchunked
- HTTP/1.1 response with invalid content-length header size declaration followed by space and null (hex '20 00')
- HTTP/1.01 declared; served chunked
- HTTP/01.1 declared; served chunked
- HTTP/2.B declared; served chunked
- HTTP/9.-1 declared; served chunked
- Double Transfer-Encoding: first empty; last chunked. Served with invalid content-length; not chunked.
- Relevant headers padded by preceding with hundreds of random custom headers

Note: Not all of these will be tested. The test house will provide a list of those being tested at the time of the testing agreement.

[R36] The test **MUST** verify that HTTP evasions as described in section 13.1 are detected.

13.1.1 Scoring Penalty

The scoring penalty for section 13.1 is 3% per missing evasion.

13.2 HTML Obfuscation

Whereas HTTP obfuscations evade detection by misusing the transmission, HTML obfuscations are contained within the content itself.

Since the following are subject to change, this should not be considered a comprehensive list but is intended to illustrate the kinds of obfuscation the test house will employ during testing.

Test Objective: This test is to determine if the SSE implementation can detect HTML obfuscations.

Test Process: The obfuscations shown below are included in HTML content and are passed to the SSE implementation under test. SSE implementations should protect against these obfuscations and others like them.

- js-binary-obfuscation
- babel-minify
- closure
- code-protect
- confusion
- jfogs
- jfogs-reverse
- jjencode
- jsbeautifier
- jsmin
- js-obfuscator
- qzx-obfuscator
- chunked and gzip compressed js-binary-obfuscation
- chunked and deflate compressed js-binary-obfuscation
- UTF-8 encoding
- UTF-8 encoding with BOM
- UTF-16 encoding with BOM
- UTF-8 encoding; no http or html declarations
- UTF-8 encoding with BOM; no http or html declarations
- UTF-16 encoding with BOM; no http or html declarations
- UTF-16-LE encoding without BOM
- UTF-16-BE encoding without BOM
- UTF-16-LE encoding without BOM; no http or html declarations
- UTF-16-BE encoding without BOM; no http or html declarations

- UTF-7 encoding
- UTF-8 encoding
- UTF-8 encoding
- EICAR string included at top of HTML
- Hex encoded script decoded using JavaScript unescape
- Unicode encoded script decoded using JavaScript unescape
- Hex encoded script as variable decoded using JavaScript unescape
- Unicode encoded script as variable decoded using JavaScript unescape
- padded with <=5MB
- padded with <=25MB
- padded with >25MB

[R37] The test **MUST** verify that HTML Obfuscation are detected.

13.2.1 Scoring Penalty

The scoring penalty for section 13.2 is 3% per missing obfuscation.

13.3 Packers

Packers are primarily used to obfuscate and "protect" compiled binaries. Along with the compressed/obfuscated data (the original binary in obfuscated form), they contain a "stub," which, upon execution, de-obfuscates the binary and jumps to its restored entry point. Malware authors typically use packing techniques to obfuscate binaries so they cannot be easily analyzed.

Test Objective: The objective of this test is for the SSE implementation to detect Packers.

Test Process: The test process is to introduce Packets and determine if they are detected by the SSE implementation. Packers including manipulation of the following attributes will be tested:

- Anskaya
- excalibur
- exefog
- Exestealth
- fearzpacker

- fishPE
- hidepx
- kkrunchy
- Krypton
- mew
- petite24
- telock
- upx
- Winkypt
- Winupack
- Yc

Since the preceding are subject to change, this should not be considered a comprehensive list but is intended to illustrate the kinds of Packers the test house will employ during testing. Vendors should protect against these Packers and others like them.

[R38] The test **MUST** verify that Malware within Packers as described in section 13.3 are detected.

13.3.1 Scoring Penalty

The scoring penalty for section 13.3 is 3% per missing Packer.

13.4 Compressors

Compressors are primarily used to reduce the size of a file. They are also used by attackers to obfuscate malware since compressed files do not look the same to pattern matching engines. As such a security product must have the matching compression algorithm in order to detect malware that has been compressed. Malware authors typically use compression techniques to obfuscate binaries so they cannot be easily analyzed.

Test Objective: This test is performed to see if the SSE implementation can detect common Compressors.

Test Process: Compressors are passed to the SSE implementation and it is determined if they are detected. Compressors including manipulation of the following attributes will be tested:

- 7zip\7zip
- 7zip\bzip2
- 7zip\gzip
- 7zip\xz
- ALZIP
- AshampooZip
- AverZip
- Bandizip
- FilZip
- Gzip
- KuaiZip
- MuZip
- PicoZip
- PowerArchiver
- QuickZip
- SimplyZip

- WinZip
- ZipitFast

Since the preceding are subject to change, this should not be considered a comprehensive list but is intended to illustrate the kinds of compressors the test house will employ during testing. Vendors should protect against these compressors and others like them. A number of the Compressors shown will be selected for the testing.

[R39] The test **MUST** verify that Malware within Compressors as described in section 13.4 are detected.

13.4.1 Scoring Penalty

The scoring penalty for section 13.4 is 3% per missing Compressor.

13.5 Layered Evasions

Test Objective: These tests determine the effectiveness of the SSE implementation when subjected to combinations of evasion techniques. This test attempts to bypass the SSE implementation by performing any legitimate combination of the previous evasion techniques.

Test Process: Multiple evasions are included in test packets delivered to the SSE Implementation. The ability of the SSE implementation to detect them is recorded.

Note: A set of 10 evasions are selected by the test house and these are used for all SSE implementations. If new evasions are determined to be applicable, the list of 10 is modified and used for all future tests. Missed Packers or Compressors will not be used here.

[R40] The test **MUST** verify that when a combination of evasion methods is used to ensure that all evasions are detected.

13.5.1 Scoring Penalty

The scoring penalty for section 13.5 is 3% for each missing evasion.

14 Testing of MEF 138 Draft (r1) Requirements

The requirements from MEF 138 r1 that are included in the following requirements. These requirements are tested by using the test methodologies defined in sections 8, 9, 10, 11, and 12.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R1	T	M	9.2.2	[R1] For each Security Function, the Security Solution **MUST** maintain a list of entries to be blocked Implicitly tested as part of tests in section 9.2.2
R2	T	M	9.2.2	[R2] For each Security Function, the Security Solution MUST allow the adding of entries to the Block List. Implicitly tested as part of tests in section 9.2.2
R3	T	M		[R3] For each Security Function, the Security Solution **MUST** allow removing a to be blocked that was added by one of the following methods: • Specified explicitly • Specified to conform to a category Implicitly tested as part of tests in section 9.2.2
R4	N			

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R5	N			
R6	N			
R7	T	M	9.2.2	[R7] For each Security Function, the Security Solution **MUST** maintain a list of entries to be allowed. Implicitly tested as part of tests in section 9.2.2
R8	T	M	9.2.2	[R8] For each Security Function, the Security Solution MUST allow the ability to add or remove entries from the list of the allowed. Implicitly tested as part of tests in section 9.2.2
R9	T	M	9.2.2	[R9] For each Security Function, the Security Solution **MUST** maintain a list of entries to be quarantined. Implicitly tested as part of tests in section 9.2.2

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R10	T	M	9.2.2	[R10] For each Security Function, the Security Solution **MUST** allow the ability to add or remove entries from the list of the quarantined. Implicitly tested as part of tests in section 9.2.2
R11	T	M	9.2.2	For each Security Function, the Security Solution **MUST** maintain a list of entries to be supported. Implicitly tested as part of tests in section 9.2.2
R12	T	M	9.2.2	[R12] For each Security Function, the Security Solution **MUST** maintain a list of entries to be unsupported. Implicitly tested as part of tests in section 9.2.2

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R13	T	M	9.2.2	[R13] The Security Solution **MUST** ensure that an entry appears on either the list of entities to be supported or unsupported but not both. Implicitly tested as part of tests in section 9.2.2
R14	T	M	9.2.2	[R14] For each Security Function, the Security Solution **MUST** ensure that each entry is on at most one of the following lists: Allow List, Block List, or Quarantine List. Implicitly tested as part of tests in section 9.2.2

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R15	T	M	9.2.2	[R15] For a given Security Function that uses lists of supported and unsupported, the Security Solution **MUST** ensure that each entry on the list of supported is also on one of the following lists for that Security Function: allow, block, quarantine. Implicitly tested as part of tests in section 9.2.2
R16	T	M	9.2.2	[R16] For a given Security Function that uses lists of supported and unsupported, the Security Solution **MUST** ensure that each entry on the list of unsupported is also on one of the following lists for that Security Function: allow, block, Quarantine. Implicitly tested as part of tests in section 9.2.2

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R17	T	M	9.2.2	[R17] For a given Security Function that uses lists of supported and unsupported, the Security Solution **MUST** ensure that each criteria entry on the list to allow is also on one of the following lists for that Security Function: supported or the unsupported. Implicitly tested as part of tests in section 9.2.2
R18	T	M	9.2.2	[R18] For a given Security Function that uses the list of supported and unsupported, the Security Solution **MUST** ensure that each entry on the list of blocked entities is also on one of the following lists for that Security Function: • supported • unsupported Implicitly tested as part of tests in section 9.2.2

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R19	T	M	9.2.2	[R19] For a given Security Function that uses the lists of supported and unsupported, the Security Solution **MUST** ensure that each entry marked as quarantine is also on one of the following lists for that Security Function: supported or unsupported. Implicitly tested as part of tests in section 9.2.2
R20	T	M	9.2.2	[R20] The Security Solution **MUST** support the use of the Notification Action Modifier, for each Security Function independently, applicable to the Security Function's Block List, Allow List, or Quarantine List. Implicitly tested as part of tests in section 9.2.2

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R21	T	M	9.2.2	[R21] All criteria entries on a Security Function's Block and Quarantine list **MUST** contain the Notification Action Modifier, unless modified by the user. Implicitly tested as part of tests in section 9.2.2
D1	T	O	9.2.2	[D1] The Security Solution **SHOULD** support the use of more than one notification level for the Notification Action Modifier. Implicitly tested as part of tests in section 9.2.2
CR1	T	O	9.2.2	[CR1] If multiple notification levels are used, the Security Solution MUST allow the timing of issuing each SEN per the notification level to be selected. Implicitly tested as part of tests in section 9.2.2
R22	N			

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R23	T	M	10.2.1	[R23] A SEN **MUST** be issued whenever a subset of a Service Flow: • matches a criteria entry that contains the Notification Action Modifier on a Security Function's Block List, Allow List, or Quarantine List, or • does not match a criteria entry on any of the Security Function's Lists but is Blocked due to the application of the Security Function Policy.
R24	N			
R25	T	M	10.2	[R25] The Security Solution **MUST** store each SEN in a secure repository for future reference and security auditing purposes.
R26	T	M	10.2	[R26] A SEN MUST include the items listed in Table 3.
D2	N			
R27	T	M	10.2	[R27] Any domain name or URL in a SEN **MUST** be neutralized.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
D3	T	O	10.2	[D3] The method for neutralizing the domain name or URL in a SEN **SHOULD** use square brackets around each period.
R28	T	M	10.2	[R28] The timestamp of the SEN **MUST** be expressed in UTC.
R29	T	M	10.2	A Security Event Notification Policy **MUST** contain the following: • List of Recipients and one or more contact methods for each recipient authorized to receive the SEN
R30	T	M	9.2.3	[R30] A SAN **MUST** be issued whenever parameter element values for a given Security Function Policy are changed.
R31	T	M	10.1	[R31] The Service Provider **MUST** store each SAN in a secure repository for future reference and auditing purposes.
R32	T	M	10.1	[R32] A SAN **MUST** include the items listed in Table 4:

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R33	T	M	10.2.1	R33 Any domain name or URL in a SAN that is associated with a Block List or Quarantine List **MUST** be neutralized.
R34	T	M	10.2.1	[R34] The timestamp of the SAN **MUST** be expressed in UTC.
R35	T	M	10.2.1	[R35] A Security Admin Notification Policy **MUST** contain the following: • List of Recipients and one or more contact methods for each recipient authorized to receive the SAN
D4	N			

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R36	T	M	8.1.1	[R36] When the Middlebox Security Function supports TLS, each Middle-box Security Function List **MUST** contain the following criteria entry parameters for TLS. ▪ Type of encryption protocol: TLS ▪ Protocol Version, e.g., 1.1, 1.2, 1.3… ▪ List of Cipher Suites for a given TLS protocol version (any can be used to indicate that any cipher suite for a given TLS protocol version is on the list)

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R37	T	M	8.1.1	[R37] When the Middlebox Security Function supports IPsec, each Middle-box Security Function List **MUST** contain the following criteria entry parameters for IPsec. ▪ Type of encryption protocol: IPsec ▪ List of Internet Key Exchange v2 (IKEv2) parameters, per Table 1 of NIST SP 800-77 Rev. 1 ▪ List of IPsec parameters, per Table 1 of NIST SP 800-77 Rev. 1

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R38	T	D		[R38] When a Middlebox Security Function Policy is included in a Security Policy for a given Service Flow, the Security solution **MUST** meet the mandatory requirements specified in Section 7.1 of MEF 138(r1) relating to the Middlebox Security Function Block List, the Middlebox Security Function Allow List, the Middlebox Security Function Supported List, and the Middlebox Security Function Unsupported List.
R39	T	D		[R39] When a Middlebox Security Function Policy is included in a Security Policy for a given Service Flow, the subset of the Service Flow that matches a criteria entry on the Middlebox Security Function Block List **MUST** be Blocked.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R40	T	D		[R40] When a Middlebox Security Function Policy is included in a Security Policy for a given Service Flow, the subset of the Service Flow that matches a criteria entry on the Middlebox Security Function Allow List and on the Middlebox Security Function Supported List **MUST** be decrypted.
R41	T	D		[R41] When a Middlebox Security Function Policy is included in a Security Policy for a given Service Flow, the subset of the Service Flow that matches a criteria entry on the Middlebox Security Function Allow List and on the Middlebox Security Function Unsupported List **MUST** be passed through the Middlebox Security Function without change.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R42	T	D		[R42] When a Middlebox Security Function Policy is included in a Security Policy for a given Service Flow, the Service Provider **MUST** support both of the following actions for a subset of the Service Flow that does not match a criteria entry on any of the Middlebox Security Function lists: ▪ Block the subset of the Service Flow ▪ Pass the subset of the Service Flow through the Middlebox Security Function without change

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R43	T	M		[R43] When a Middlebox Security Function Policy is included in a Security Policy for a given Service Flow, the Middlebox Security Function **MUST** perform one of the following actions for each subset of the Service Flow that does not match a criteria entry on any of the Middlebox Security Function lists: ▪ Block the subset of the Service Flow ▪ Pass the subset of the Service Flow through the Middlebox Security Function without change
D5	N			
R44	T	M		The Middlebox Security Function **MUST** support at least one of the following secure transport protocols: ▪ Transport Layer Security (TLS) ▪ IPsec
R45	N			
R46	N			

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R47	T	M		[R47] When TLS is supported, the Middlebox Security Function **MUST** meet the mandatory requirements of TLS 1.2, per RFC 5246.
R48	T	M		[R48] When TLS is supported, the Middlebox Security Function **MUST** meet the mandatory requirements of Section 9.3 of RFC 8446 (Protocol Invariants).
R49	T	M		[R49] When IPsec is supported, the Middlebox Security Function **MUST** meet the mandatory requirements of IKEv2, per RFC 7296.
R50	T	M		When a Middlebox Security Function rules (policy) is included in a Security rules (policy) for a given Service Flow, the Middlebox Security Function **MUST NOT** change the protocol implementation or cryptographic suite of the session as compared to the client (TLS) or initiator endpoint (IPsec) request.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R51	T	M		When a Middlebox Security Function rules (policy) is included in a Security rules (policy) for a given Service Flow, the Middlebox Security Function **MUST NOT** choose a weaker cipher suite in the negotiation for a session as compared to the session without the Middlebox Security Function.
R52	T	M		[R52] A Service Flow decrypted by the Middlebox Security Function **MUST NOT** be exposed outside the Security Solution's Security Functions in an unencrypted form or in an encrypted form that offers a lower level of confidentiality and integrity than the originally encrypted Service Flow.
R53	T	M		[R53] When TLS or other protocols using PKI are used, the Middlebox Security Function **MUST** be capable of issuing valid, signed certificates for each encrypted session.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R54	T	M		[R54] When TLS or other protocols using PKI are not used, the Middlebox Security Function **MUST** be capable of issuing Key Material for each encrypted session.
R55	T	M		[R55] When TLS or other protocols using PKI are used and when performing inspection within the Middlebox Security Function, it **MUST** be possible to use certificates that are backed by a CA where the trust path and issuer can be validated by users who have installed the full certificate chain on their computer.
R56	N			
R57	T	M		[R57] Server certificates and/or other Key Material **MUST** be generated and regenerated with fresh, suitably random material per the requirements in FIPS- 140-2 for which the Middlebox Security Function processes Service Flows.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R58	T	M		[R58] All replacement properties for each encrypted session, e.g., alternative certificate server names, certificate validity periods, and choices of cipher suites **MUST NOT** reduce the level of security functionality.
R59	T	M		[R59] Where a CA is operated in support of the inspection within the Middlebox Security Function, it **MUST** clearly identify itself within the visible issuer properties (for TLS, within the certificates, as defined in Section 4.1.2.4 of RFC 5280) presented, for reasons of transparency, so that the user can identify when a Middlebox Security Function is in the path versus when they are connecting directly to the originating server.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
D6	T	D		[D6] Any TLS based interception being performed by a Middlebox Security Function as part of a managed service **SHOULD** use a Public Key Infrastructure (PKI) hierarchy that is rooted in a CA that is operated in line with the CA/Browser Forum baseline requirements where certificates are securely created, used, revoked and destroyed
D7	T	D		[D7] Where possible, CAs **SHOULD** log all certificates that they issue using the standardized Certificate Transparency (CT) security standard, see RFC 6962.
R60	T	M		[R60] The Middlebox Security Function **MUST** be capable of accepting a valid, signed Subscriber certificate for each TLS session between the Middlebox Security Function and the user's server.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R61	T	M		[R61] When a Middlebox Security Function rule (policy) is included in a Security rule (policy) for a given Service Flow, the Middlebox Security Function **MUST** verify the validity/identity of the client and server (TLS), as illustrated in Figure 3 of MEF 138 (r1).
R62	T	M		[R62] When a Middlebox Security Function rule (policy) is included in a Security rules (policy) for a given Service Flow, the Middlebox Security Function **MUST** verify the validity/identity of the initiator and responder endpoints (IPsec).
R63	T	M		[R63] When an invalid/unknown client and server (TLS) is detected, the Middlebox Security Function **MUST** be capable of performing a variety of behaviors including blocking, allowing, etc., and notifying the client (TLS). .

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R64	T	M		[R64] When an invalid/unknown initiator and responder endpoint (IPsec), is detected, the Middlebox Security Function MUST be capable of performing a variety of behaviors including blocking, allowing, etc., and notifying the initiator endpoint (IPsec) of the discrepancy.
R65	T	M		Each criteria entry on an IP, Port and Protocol Filtering List **MUST** include the fields listed in Table 5 of MEF 138 (r1).
R66	T	M		[R66] When an IP, Port and Protocol Filtering Security Function rule (policy) is included in a set of Security rules (policy) for a given Service Flow, the Service Provider **MUST** meet the mandatory requirements specified in Section 7.1 of MEF 138 (r1) relating to the IP, Port and Protocol Filtering Block List, the IP, Port and Protocol Filtering Allow List and the IP, Port and Protocol Filtering Quarantine List.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R67	T	M	9.2.2	[R67] When an IP, Port and Protocol Filtering Security Function Policy is included in a Security Policy for a given Service Flow, the Security Solution **MUST** support both of the following actions for a subset of the Service Flow that does not match a criteria entry on any of the IP, Port and Protocol Filtering lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R68	T	M	9.2.2	[R68] When an IP, Port and Protocol Filtering Security Function rules (policy) is included in a Security rule (policy) for a given Service Flow, the IP, Port and Protocol Filtering Security Function **MUST** perform one of the following actions for each subset of the Service Flow that does not match a criteria entry on any of the IP, Port and Protocol Filtering lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

R69	T	M	9.2.2	When an IP, Port and Protocol Filtering Security Function rule (policy) is included in a Security rule (policy) for a given Service Flow, the IP, Port and Protocol Filtering Security Function **MUST** perform one of the following actions: ▪ Allow the subset of the Service Flow that matches a criteria entry on the IP, Port and Protocol Filtering Allow List ▪ Allow the subset of the Service Flow that does not match a criteria entry on any of the IP, Port and Protocol Filtering lists, per the second bullet of [R68] in MEF 138 Draft (r1) ▪ Block the subset of the Service Flow that matches a criteria entry on the IP, Port and Protocol Filtering Block List ▪ Block the subset of the Service Flow that matches a criteria entry on the IP, Port and Protocol Filtering Quarantine List ▪ Block the subset of the Service Flow that does not match a criteria entry

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
				on any of the IP, Port and Protocol Filtering lists, per the first bullet of [R68] in MEF 138 Draft (r1)

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
D8	T	D		[D8] The IP, Port and Protocol Filtering Security Function **SHOULD** use the same set of Block/Allow/Quarantine Lists for Service Flows including DNS messages and Service Flows not including DNS messages.
R70	T	D		{R70] The Domain Name Filtering Security Function **MUST** be capable of using wildcard criteria entries.
R71	N			
R72	T	D		When a Domain Name Filtering Security Function Policy is included in a Security Policy for a given Service Flow, the Security solution **MUST** meet the mandatory requirements specified in Section 7.1 of MEF 138 Draft (r1) relating to the Domain Name Filtering Block List, the Domain Name Filtering Allow List and the Domain Name Filtering Quarantine List.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R73	T	D		[R73] When a Domain Name Filtering Security Function Policy is included in a Security Policy for a given Service Flow, the Security solution **MUST** support both of the following actions for a subset of the Service Flow that does not match a criteria entry on any of the Domain Name Filtering lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R74	T	D		[R74] When a Domain Name Filtering Security Function rule (policy) is included in a Security rule (policy) for a given Service Flow, the Domain Name Filtering Security Function **MUST** perform one of the following actions for each subset of the Service Flow that does not match a criteria entry on any of the Domain Name Filtering lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

R75	T	D		When a Domain Name Filtering Security Function rule (policy) is included in a Security rule (policy) for a given Service Flow, the Domain Name Filtering Security Function **MUST** perform one of the following actions: ▪ Allow the subset of the Service Flow that matches a criteria entry on the Domain Name Filtering Allow List ▪ Allow the subset of the Service Flow that does not match a criteria entry on any of the Domain Name Filtering lists, per the second bullet of [R74] of MEF 138 Draft (r1) ▪ Block the subset of the Service Flow that matches a criteria entry on the Domain Name Filtering Block List ▪ Block the subset of the Service Flow that matches a criteria entry on the Domain Name Filtering Quarantine List ▪ Block the subset of the Service Flow that does not match a criteria entry on any of the Domain Name Filtering lists, per the first bullet

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
				of [R74] of MEF 138 Draft (r1)

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
D9	T	D		[D9] The Domain Name Filtering Security Function **SHOULD** use the same set of Block/Allow/Quarantine Lists for Service Flows including DNS messages and Service Flows not including DNS messages.
R76	T	M	9.2.2	[R76] The URL Filtering Security Function **MUST** be capable of using wildcard criteria entries.
R77	N			
R78	T	M	9.2.2	[R78] When a URL Filtering Security Function rule (policy) is included in a Security rules (policy) for a given Service Flow, the ZT solution **MUST** meet the mandatory requirements specified in Section 7.1 of MEF 138 Draft (r1) relating to the URL Filtering Block List, the URL Filtering Allow List, and the URL Filtering Quarantine List.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R79	T	M	9.2.2	[R79] When a URL Filtering Security Function rule (policy) is included in a Security rules (policy) for a given Service Flow, the ZT solution **MUST** support both of the following actions for a subset of the Service Flow that does not match a criteria entry on any of the URL Filtering lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow
R80	T	M	9.2.2	When a URL Filtering Security Function rule (policy) is included in a Security rules (policy) for a given Service Flow, the URL Filtering Security Function **MUST** perform one of the following actions for each subset of the Service Flow that does not match a criteria entry on any of the URL Filtering lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

R81	T	M	9.2.2	[R81] When a URL Filtering Security Function rules (policy) is included in a Security rules (policy) for a given Service Flow, the URL Filtering Security Function **MUST** perform one of the following actions: ▪ Allow the subset of the Service Flow that matches a criteria entry that is on the URL Filtering Allow List ▪ Allow the subset of the Service Flow that does not match a criteria entry on any of the URL Filtering lists, per the second bullet of [R80] in MEF 138 Draft (r1) ▪ Block the subset of the Service Flow that matches a criteria entry on the URL Filtering Block List ▪ Block the subset of the Service Flow that matches a criteria entry on the URL Filtering Quarantine List ▪ Block the subset of the Service Flow that does not match a criteria entry on any of the URL Filtering lists, per the first bullet of [R80] in MEF 138 Draft (r1)

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R82	T	M	10.2.3	When a Malware Detection and Removal Security Function rules (policy) are included in a Security rules (policy) for a given Service Flow, the Security solution **MUST** meet the mandatory requirements specified in Section 7.1 of MEF 138 Draft R1 relating to the Malware Detection and Removal Block List, the Malware Detection and Removal Allow List and the Malware Detection and Removal Quarantine List.
R83	T	M	10.2.3	[R83] When a Malware Detection and Removal Security Function rule (policy) is included in a Security rules (policy) for a given Service Flow, the Security solution **MUST** describe which kind of detection (e.g., signature scan, behavioral analysis, or both) is performed.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R84	T		9.2.2	When a Malware Detection and Removal Security Function is included in a Service rules (policy) for a given Service Flow, the Security solution **MUST** support both of the following actions for a subset of the Service Flow that does not match a criteria entry on any of the Malware Detection and Removal Filtering lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R85	T	M	9.2.2	[R85] When a Malware Detection and Removal Security Function is included in a Service rule (policy) for a given Service Flow, the Malware Detection and Removal Security Function **MUST** perform one of the following actions for each subset of the Service Flow that does not match a criteria entry on any of the Malware Detection and Removal lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R86	T	M	9.2.2	[R86] When a Malware Detection and Removal Security Function is included in a Service rule (policy) for a given Service Flow, and when a subset of the Service Flow is determined to either have Malware or look suspicious that it may have Malware, the Malware Detection and Removal Security Function **MUST** perform one of the following actions: • Block the Service Flow • Block the subset of the Service Flow containing the Malware and Allow the remainder of the Service Flow • Quarantine the Service Flow • Quarantine the subset of the Service Flow containing the Malware and Allow the remainder of the Service Flow • Remove Malware from the Service Flow and Allow the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R87	T	M	10.2.3	[R87] If a SEN is issued ([R23]) of MEF 138 Draft (r1), the Security solution **MUST** report which action is taken for detected Malware and make it available via the SEN (see MEF 138 Draft (r1) Section 7.2).
R88	T	D		When a Data Loss Prevention Security Function is included in a Service rules (policy), the Security solution **MUST** meet the mandatory requirements specified in MEF 138 Draft (r1) Section 7.1 relating to the Data Loss Prevention Block List, the Data Loss Prevention Allow List, and the Data Loss Prevention Quarantine List.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R89	T	D		[R89] When a Data Loss Prevention Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, the Security solution **MUST** support both of the following actions for a subset of the Service Flow that does not match a criteria entry on any of the Data Loss Prevention lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R90	T	D		[R90] When a Data Loss Prevention Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, the Data Loss Prevention Security Function **MUST** perform one of the following actions for each subset of the Service Flow that does not match a criteria entry on any of the Data Loss Prevention lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

R91	T	D		[R91] When a Data Loss Prevention Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, and when a subset of the Service Flow is determined to either have PII or CPI, the Data Loss Prevention Security Function **MUST** perform one of the following actions: ▪ Allow the subset of the Service Flow that matches a criteria entry that is on the Data Loss Prevention Allow List ▪ Allow the subset of the Service Flow that does not match a criteria entry on any of the Data Loss Prevention lists, per the second bullet of [R90] of MEF 138 Draft (r1) ▪ Block the Service Flow that matches a criteria entry on the Data Loss Prevention Block List ▪ Block the subset of the Service Flow that matches a criteria entry on the Data Loss Prevention Block List and Allow the remainder of the Service Flow ▪ Block the subset of the Service Flow that matches a criteria entry

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
				on the Data Loss Prevention Quarantine List and Allow the remainder of the Service Flow ▪ Block the subset of the Service Flow that does not match a criteria entry on any of the Data Loss Prevention lists, per the first bullet of MEF 138 Draft (r1) [R90] ▪ Remove the PII and/or CPI from the subset of the Service Flow that matches a criteria entry on either the Data Loss Prevention Block List or the Data Loss Prevention Quarantine List and Allow the remainder of the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R92	T	D		[R92] When a Data Loss Prevention Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, the Security solution **MUST** support both of the following actions for a subset of the Service Flow that cannot be scanned: • Block the subset of the Service Flow • Allow the subset of the Service Flow
R93	T	D		[R93] When a Data Loss Prevention Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, the Data Loss Prevention Security Function **MUST** perform one of the following actions for a subset of the Service Flow that cannot be scanned: • Block the subset of the Service Flow • Allow the subset of the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R94	T	D		[R94] Each criteria entry on a DNS Protocol Filtering List **MUST** include the fields listed in Table 6 of MEF 138 Draft (r1).
R95	T	D		[R95] When a DNS Protocol Filtering Security Function rule (policy) is included in a Security rules (policy) for a given Service Flow, the Service Provider **MUST** meet the mandatory requirements specified in Section 7.1 of MEF 138 Draft (r1) relating to the DNS Protocol Filtering Block List, the DNS Protocol Filtering Allow List and the DNS Protocol Filtering Quarantine List.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R96	T	D		[R96] When a DNS Protocol Filtering Security Function rule (policy) is included in a Security riles (policy) for a given Service Flow, the Security solution **MUST** support both of the following actions for a subset of the Service Flow that does not match a criteria entry on any of the DNS Protocol Filtering lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R97	T	D		[R97] When a DNS Protocol Filtering Security Function rule (policy) is included in a Security rule (policy) for a given Service Flow, the DNS Protocol Filtering Security Function **MUST** perform one of the following actions for each subset of the Service Flow that does not match a criteria entry on any of the DNS Protocol Filtering lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

R98	T	D		[R98] When a DNS Protocol Filtering Security Function rule (policy) is included in a Security rule (policy) for a given Service Flow, the DNS Protocol Filtering Security Function **MUST** perform one of the following actions, based on agreement between the Service Provider and the Subscriber: ▪ Allow the subset of the Service Flow that matches a criteria entry on the DNS Protocol Filtering Allow List ▪ Allow the subset of the Service Flow that does not match a criteria entry on any of the DNS Protocol Filtering lists, per the second bullet of [R97] in MEF 138 Draft (r1) ▪ Block the subset of the Service Flow that matches a criteria entry on the DNS Protocol Filtering Block List ▪ Block the subset of the Service Flow that matches a criteria entry on the DNS Protocol Filtering Quarantine List ▪ Block the subset of the Service Flow that does not match a criteria entry

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
				on any of the DNS Protocol Filtering lists, per the first bullet of [R97] in MEF 138 Draft (r1)

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R99	T	D		[R99] The DNS Protocol Filtering Security Function **MUST** be capable of informing the DNS client immediately of any DNS message failure.
D10	T	D		[D10] When a DNS Protocol Filtering Security Function rule (policy) is included in a Security rule (policy) for a given Service Flow, and when a DNS message in that Service Flow is Blocked, the DNS Protocol Filtering Security Function **SHOULD** send an appropriate DNS response code, per Section 2.3 of RFC 6895 to the DNS client.
R100	T	D		[R100] Implementations **MUST** be capable of inspecting and influencing DNS either directly or indirectly even before any future Service Flow is established.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R101	T	D		[R101] The contents of DNS requests from clients and the resultant responses from DNS servers **MUST** themselves be evaluated in the same manner as other Service Flows.

R102	T	D		[R102] When a Protective DNS Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, the Security solution **MUST** support the ability to perform resolution for the following DNS resource record types. • SOA (Start of Authority) – specifies authoritative information about a DNS zone, including the primary name server, the email of the domain administrator, the domain serial number, and several timers relating to refreshing the zone (RFC 1035) • NS (Name Server) – delegates a DNS zone to use the given authoritative name servers (RFC 1035) • A – Returns a 32-bit IPv4 address, most commonly used to map hostnames to an IP address of the host, but it is also used for DNSBLs, storing subnet masks in RFC 1101, etc. (RFC 1035) • AAAA – Returns a 128-bit IPv6 address, most commonly used to map hostnames to an IP

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
				address of the host (RFC 3596) • CNAME (Canonical Name) – alias of one name to another: the DNS lookup will continue by retrying the lookup with the new name (RFC 1035) • MX (Mail Exchange) – maps a domain name to a list of message transfer agents for that domain (RFC 1035) • SRV (Service) – delegates a DNS zone to use the given authoritative name servers for the purpose of service lookups, e.g., SIP, XMPP (RFC 2782) • PTR (Pointer) – pointer to a canonical name. Unlike a CNAME, DNS processing stops and just the name is returned. The most common use is for implementing reverse DNS lookups, but other uses include such things as DNS-SD (RFC 6763)

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
D11	T	D		[D11] When a Protective DNS Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, the Security solution **SHOULD** support the ability to perform resolution for the following DNS resource record types.
R103	T	M		[R103] For records where resolution of the record type is intended to be limited, inspection of responses **MUST** at minimum consist of comparing the record value (by IP and/or hostname) using the same source of truth, and with the same rules that would otherwise be applied via the traffic inspection within the Middlebox Security Function.
R104	T	D		[R104] Each criteria entry on a Protective DNS List **MUST** include the fields listed in Table 7 of MEF 138 Draft (r1) for the appropriate DNS resource record type identified in [R102] in MEF 138 Draft (r1)

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
D12	T	D		[D12] A Protective DNS Security Function **SHOULD** support the ability to inspect DNS messages encrypted with DNS over HTTPS (DoH), per RFC 8484
D13	T	D		[D13] A Protective DNS Security Function **SHOULD** support the ability to inspect DNS messages encrypted with DNS over TLS (DoT), per RFC 7858
D14	T	D		[D14] A Protective DNS Security Function **SHOULD** support the ability to inspect DNS messages encrypted with DNSSEC, per RFCs 4033, 4034, and 4035

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R105	T	D		[R105] When a Protective DNS Security Function is included in a Service rule (policy), the Security solution **MUST** meet the mandatory requirements specified in Section 7.1 of MEF 138 Draft (r1) relating to the Protective DNS Block List, the Protective DNS Allow List, the Protective DNS Quarantine List, the Protective DNS Supported List, and the Protective DNS Unsupported List.
R106	T	D		[R106] When a Protective DNS Security Function rule (policy) is included in a Security rules (policy) for a given Service Flow, the subset of the Service Flow that matches a criteria entry on the Protective DNS Security Function Block List **MUST** be Blocked.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R107	T	D		[R107] When a Protective DNS Security Function rule (policy) is included in a Security rule (policy) for a given Service Flow, the subset of the Service Flow that matches a criteria entry on the Protective DNS Security Function Allow List and on the Protective DNS Security Function Supported List **MUST** be processed.
R108	T	D		[R108] When a Protective DNS Security Function rule (policy) is included in a Security rule (policy) for a given Service Flow, the subset of the Service Flow that matches a criteria entry on the Protective DNS Security Function Allow List and on the Protective DNS Security Function Unsupported List **MUST** be passed through the Protective DNS Security Function without change.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R109	T	D		[R109] When a Protective DNS Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, the Security solution **MUST** support both of the following actions for a subset of the Service Flow that does not match a criteria entry on any of the Protective DNS lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R110	T	D		[R110] When a Protective DNS Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, the Protective DNS Security Function **MUST** perform one of the following actions for each subset of the Service Flow that does not match a criteria entry on any of the Protective DNS lists: • Block the subset of the Service Flow • Allow the subset of the Service Flow

R111	T	D		[R111] When a Protective DNS Security Function rule (policy) is included in a Service rule (policy) for a given Service Flow, and when a subset of the Service Flow is transporting DNS messages, the Protective DNS Security Function **MUST** perform one of the following actions: • Allow the DNS message that matches a criteria entry that is on the Protective DNS Allow List • Allow the DNS message that does not match a criteria entry on any of the Protective DNS lists, per the second bullet of [R110] • Block the DNS message that matches a criteria entry on the Protective DNS Block List • Block the DNS message that matches a criteria entry on the Protective DNS Quarantine List • Block the DNS message that does not match a criteria entry on any of the Protective DNS lists, per the first

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
				bullet of [R110] in MEF 138 Draft (r1) • Alter the DNS message response to protect the recipient

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R112	T	D		[R112] The Protective DNS Security Function **MUST** be capable of informing immediately when a DNS message is either Blocked or altered.
R113	N			[R113] When a Security Function rule (policy) is in force for Service Flows, the Security solution **MUST** inform of any expected impact to the service performance metrics
R114	T	D		[R114] When a Service uses a Security rule (policy), the Security rule (policy) Identifier **MUST** be an Identifier String.
R115	T	D		[R115] When a Service uses a Security rule (policy), each Security rule (policy) Identifier **MUST** be unique among all Security rules (policies) for a given Service instance.

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R116	T	D		[R116] The Security Function rule (policy) Identifier **MUST** be an Identifier String
R117	T	D		[R117] Each Security Function rule (policy) Identifier **MUST** be unique among all Security Function rules (policies) for a given Service instance.
R118	T	D		[R118] Each Middlebox Security Function rule (policy) **MUST** contain the parameters listed in Table 8 of MEF 138 Draft (r1).
R119	T	D		[R119] Each IP, Port and Protocol Filtering Security Function rule (policy) **MUST** contain the parameters listed in Table 9 of MEF 138 Draft (r1).
R120	T	D		[R120] Each Domain Name Filtering Security Function rule (policy) **MUST** contain the parameters listed in Table 10 of MEF 138 Draft (r1).

MEF 138 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R121	T	D		[R121] Each URL Filtering Security Function rule (policy) **MUST** contain the parameters listed in Table 11 of MEF 138 Draft (r1).
R122	T	D		[R122] Each Malware Detection and Removal Security Function rule (policy) **MUST** contain the parameters listed in Table 12 of MEF 138 Draft (r1).
R123	T	D		[R123] Each Data Loss Prevention Security Function rule (policy) **MUST** contain the parameters listed in Table 13 of MEF 138 Draft (r1).
R124	T	D		Each DNS Protocol Filtering Security Function rule (policy) **MUST** contain the parameters listed in Table 14 of MEF 138 Draft (r1).
R125	T	D		Each Protective DNS Security Function rule (policy) **MUST** contain the parameters listed in Table 15 of MEF 138 Draft (r1).

Table 9 – MEF 138 Testable Requirements

15 Testing of MEF 117 Requirements

The focus of this section is to identify which requirements from MEF 117 [10] are tested using the test methodologies defined in sections 8, 9, 10, 11, and 12.

Requirements are either Mandatory, meaning that they must be tested as a part of certification, Optional, meaning that they may be tested as a part of certification, or Deferred, meaning that they are requirements focused on Service Providers who are not included in the scope of this certification test.

MEF 117 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R1	N			
R2	N			
R3	N			
R4	N			
R5	N			
R6	N			
R7	N			
R8	N			
R9	N			
R10	N			
R11	N			
R12	N			
R13	N			
R14	N			
R15	N			
R16	N			
R17	N			
R18	N			
R19	N			
R20	N			
R21	N			
R22	N			
R23	N			
D1	N			
R24	N			
R25	N			
R26	N			
R27	N			
R28	T	D		Revised Requirement [R28] Each SSE **MUST** contain at least one SSE rule (policy) End Point.
R29	N			ZT requirement
R30	N			ZT requirement
R31	N			ZT requirement
R32	N			ZT requirement
D2	N			ZT requirement
R33	N			ZT requirement

MEF 117 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R34	N			ZT requirement
R35	N			ZT requirement
D3	N			ZT requirement
R36	N			
R37	N			
R38	N			ZT requirement
R39	N			ZT requirement
R40	N			ZT requirement
R41	N			
R42	N			
R43	T	M		[R43] The SSE **MUST** support UTC for any timestamps. Need test case
D4	T	O		[D4] The SSE **SHOULD** support setting the time zone for an SSE implementation. Need test case
D5	N			See MEF W90.2
R44	N			
R45	N			

MEF 117 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R46	T	M	11.3, 11.4	[R46] A SSE Service **MUST** support the following Security Functions to be specified in the Security rule (policy) associated with Sessions: • Middle Box Function (MBF) (deferred) • IP, Port and Protocol Filtering (IPPF) (deferred) • DNS Protocol Filtering (DPF) (deferred) • Domain Name Filtering (DNF) (deferred) • URL Filtering (URLF) (deferred) • Malware Detection and Removal (MD+R)
R47	N			SP Requirement
R48	N			SP Requirement
R49	T	D		[R49] Each SSE Security Function Atomic Policy **MUST** utilize the appropriate parameters for Security Functions as defined in MEF 138 Draft (r1) section 9.
R50	N			ZT requirement
R51	N			SP Requirement
R52	N			SP Requirement
R53	N			SP Requirement
R54	N			SP Requirement
R55	N			SP Requirement

MEF 117 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
R56	N			SP Requirement
R57	N			SP Requirement
D6	N			SP Requirement
R58	N			
R59	N			
R60	N			
R61	T	M	10.2	[R61} An SSE implementation **MUST** include the following rules (policy): • Security rules (policy).
R62	T	M	9.2	[R62] An SSE rules (policy) **MUST** have a unique identifier.
R63	N			SP requirement
R64	N			SP Requirement
R65	N			ZT Requirement
R66	N			ZT requirement
R67	N			ZT requirement
R68	N			ZT requirement
R69	N			ZT requirement
R70	N			ZT requirement
D7	N			ZT requirement
R71	T	M	9.1, 9.2, 9.3	[R71] The SSE implementation **MUST** support the use of TLS 1.2.

MEF 117 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
D8	N			[D8] The SSE implementation **SHOULD** support the use of IPSEC Subject User, Device, or Application Access Connection. This should apply to SASE testing (MEF W166)
R72	T	M	11.1, 11.2.1, 11.2.2, 11.2.3, 11.2.4, 11.3	[R72] The SSE implementation **MUST** meet the mandatory requirements of TLS 1.2, per RFC 5246 .
R73	T	M	11.1, 11.2.1, 11.2.2, 11.2.3, 11.2.4, 11.3	[R73] The SSE implementation **MUST** meet the mandatory requirements of RFC 8446 (TLS 1.3) section 9.3 (Protocol Invariants).
R74	N			SP Requirement
R75	N			SP Requirement
CR1	N			
CR2	N			
R76	N			ZT requirement
R77	N			
R78	N			ZT requirement
R79	N			
R80	N			
R81	N			
R82	N			ZT requirement
R83	N			SP Requirement
R84	N			
R85	N			ZT requirement

MEF 117 Requirement	Testable (T) or Not Testable (N)	Mandatory (M), Deferred (D), or Optional (O)	Test Methodology	Comments
CR3	N			SD-WAN requirement
CR4	N			SD-WAN requirement
R86	N			
R87	N			ZT requirement
R88	N			
R89	N			ZT requirement
R90	N			SP Requirement
R91	N			SP Requirement
R92	N			SP Requirement
D9	N			SP Requirement
D10	N			SP Requirement

Table 10 – MEF 117 Requirements

Editor Note 13: The test methodology is still being discussed for rows that have a test methodology of Needs Test Methodology. Once finalized, the test methodology will be updated.

16 Rating Methodology

The method used to determine the rating for an SD-WAN Edge Vendor solution or a SP SWVC solution under test use objective methods to provide a rating. Ratings use a 0-to-800-point scale. The point values for each rating are shown in Table 11.

Rating	Minimum Points	Maximum Points
AAA	775	800
AA	720	774
A	660	719
BBB	590	659
BB	540	589
B	480	539
CCC	420	479
CC	360	419
C	300	359
D	0	299

Table 11 – Rating Point Values

Each session of testing begins with the allocation of 800 points. Points are then deducted from the 800 points when a test does not perform as specified. A percentage of points is allocated to specific sections of the document.

Section Number	Total Points	Penalty	Comments
8.1.1		No penalty	Record values
8.1.2		No penalty	Record values
8.2.1		No penalty	Record values
8.2.2		No penalty	Record values
8.3.1.1		No penalty	Record values
8.3.1.2		No penalty	Record values
8.3.1.3		No penalty	Record values
8.3.1.4		No penalty	Record values
8.3.1.5		No penalty	Record values
8.3.2.1		No penalty	Record values
8.3.2.2		No penalty	Record values
8.3.2.3		No penalty	Record values
8.3.2.4		No penalty	Record values
8.3.2.5		No penalty	Record values
8.3.3.1		No penalty	Record values
8.3.3.2		No penalty	Record values
8.3.3.3		No penalty	Record values
8.3.3.4		No penalty	Record values
8.3.3.5		No penalty	Record values
8.3.4.1		No penalty	Record values
8.3.4.2		No penalty	Record values
8.3.4.3		No penalty	Record values
8.3.4.4		No penalty	Record values
8.3.4.5		No penalty	Record values
	0		
9.1.1		100%	
9.1.2		100%	
9.2		100%	
9.3		100%	
	800		
10.1.1		100%	
10.2.1.1		100%	
10.2.1.2		100%	
10.2.1.3		100%	
10.2.1.4		100%	
10.2.1.5		100%	
10.2.1.6		100%	
10.2.2.1		100%	
10.2.2.2		100%	
10.2.2.3		100%	

Section Number	Total Points	Penalty	Comments
10.2.2.2.2		100%	
10.2.2.2.3		100%	
10.2.2.2.4		100%	
10.2.3		25%	
	800		
11.1		50%	
11.2.1		100%	
11.2.2		10%	
11.2.3		10%	
11.2.4		5%	
11.3		15%	
	800		
12.1.1		100%	
12.1.2		100%	
12.2		100%	
12.3		10% per missing exploit	
12.3.2		100%	
12.3.3		100%	
12.3.4		100%	
12.3.5		100%	
12.4		50%	
	800		
13.1		3% per missing evasion	
13.2		3% per missing evasion	
13.3		3% per missing evasion	
13.4		3% per missing evasion	
13.5		3% per missing evasion	
	800		
Total Points			

Table 12 – Point Penalty Allocation per Section

As can be seen in Table 12, some areas of testing are considered "table stakes" for an SSE Vendor solution and test results that indicate that the expected capabilities are not provided results in a significant penalty.

Other areas of testing are considered "nice to have" functions and a lower penalty is deducted if the test results in these areas are lower than expected.

Some areas of testing are used as benchmarks for performance of the SD-WAN Edge Vendor solution or a SP SWVC solution that are compared to other tests that introduce impairments or high loads to determine their impact on the performance. No points are deducted for these areas.

The percentage of penalty is calculated and deducted from the total points and the total points associated with the testing are determined. The overall rating is determined based on the total points as shown in Table 11. Scores are not provided for each section of the testing, only for the overall results.

Note: If a specific exploit, malware, or evasion fails an initial test it is not used again when steps like layered evasions are tested.

16.1 MEF Certification Pass/Fail Criteria

To allow for a MEF Certification a Pass/Fail criteria has been defined within this section. Scores are calculated as describe below.

It is proposed that a minimum of 90% of the requirements from MEF 88 and MEF 117 shown in section 14 (MEF 88) and section 15 (MEF 117) as testable are required to pass in order for an SSE solution to be eligible for MEF Certification. Solutions that fail to meet these criteria may still be provided a rating, but need are not MEF certified.

17 References

[1] IETF RFC 1918, *Address Allocation for Private Internets*, by Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, February 1996

[2] IETF RFC 1951, *DEFLATE Compressed Data Format Specification version 1.3*, by P. Deutsch, May 1996, Copyright (c) 1996 L. Peter Deutsch

[3] IETF RFC 1952, *GZIP file format specification version 4.3*, by P. Deutsch, May 1996, Copyright (c) 1996 L. Peter Deutsch

[4] IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*, by Scott Bradner, March 1997

[5] IETF RFC 2616, *Hypertext Transfer Protocol -- HTTP/1.1*, by R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, June 1999, Copyright (C) The Internet Society (1999). All Rights Reserved.

[6] IETF RFC 7932, *Brotli Compressed Data Format*, by J. Alakuijala, Z. Szabadka, July 2016, Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

[7] IETF RFC 8174, *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*, by B Leiba, May 2017, Copyright © 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

[8] MEF 88, *Application Flow Security for SD-WAN Services*, November 2021

[9] MEF W90.2, *SD-WAN Certification Phase 2*, April 2023

[10] MEF 117, *SASE Service Attributes and Framework*, October 2022

[11] MEF 118, *Zero Trust Framework for MEF Services*, October 2022

[12] MEF W138, *Security Functions for IP Services*, August 2023