Requirements and Recommendations for MAEC Compatibility

Document version: 1.1 **Date:** July 7, 2013

Authors:

Robert A. Martin, MAEC Compatibility Lead — ramartin@mitre.org
Penny Chase, MAEC Technical Lead — pc@mitre.org
Ivan Kirillov, MAEC Architect — ikirillov@mitre.org
Desiree Beck, MAEC Contributor — dbeck@mitre.org

Table of Contents

Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in *RFC 2119*[1].

[1] RFC 2119 – Key words for use in RFCs to Indicate Requirement Levels; http://www.ietf.org/rfc/rfc2119.txt

Definitions

Capability —

A specific function or functions of a Tool, Service, or Repository.

Content —

Any form of MAEC entity, including MAEC Output Format documents as well as embedded elements and types.

Correctness Testing —

The process of determining whether a Tool has correctly implemented MAEC.

MAEC Bundle —

A standard form of MAEC output to capture all of the analysis-derived characteristics for a single Malware Instance, including any observed MAEC Behaviors or Actions, and any related MAEC Objects.

MAEC Container —

A standard form of MAEC output to capture one or more MAEC Packages.

MAEC Output Format —

Any of the three standard forms of MAEC output, including the MAEC Container, Package, or Bundle.

MAEC Package —

A standard form of MAEC output to characterize all known data for one or more Malware Subjects, including their analysis derived characteristics (via MAEC Bundles) and any associated analysis or other metadata.

Malware Element —

A behavior, attribute, exploit, payload, etc. that is related to a specific Malware Instance or to a family or class of Malware Instances.

Malware Instance —

A specific copy of malware.

Malware Pattern —

An abstraction of some attributes common to a set of Malware Instances (families or classes). A single Malware Pattern may potentially have many varying Malware Instances associable with it.

Malware Subject —

A MAEC entity that captures all the details pertaining to a single Malware Instance, including any corresponding analysis metadata, analysis content, and relationship information.

Capability Owner —

The custodian (real person or company) having responsibility for the Capability.

Repository —

An implicit or explicit collection of Malware Elements or Malware Patterns that supports a Content Creation Tool or Service, e.g., a database of behavioral patterns, the set of Malware Instances analyzed by a Sandbox Tool, or the aggregate output of a Static or Dynamic Binary Analysis Tool. A Repository can also be a collection of MAEC Output Format documents.

Review —

The process of determining whether a Capability is MAEC-compatible.

Review Authority —

An entity that performs a review. (The MITRE Corporation is the only review authority at this time).

Review Sample —

A copy of a Capability's output provided to a Review Authority for use in determining whether the Capability is MAEC-compatible.

Review Version —

The dated version of MAEC that is being used for determining MAEC compatibility of a Capability.

Service —

A malware analysis, detection, or remediation activity that implements one or more Capabilities.

Capability Test Results —

The dataset that represents the outcome of Correctness Testing.

Tool —

A software application or device that implements one or more functionalities. A Tool analyzes, detects, or remediates malware through various methods, e.g., a static analysis tool, dynamic analysis tool, etc. A Tool can also perform Content authoring.

User —

A consumer (or potential consumer) of the Capability.

Compatibility Capabilities

The MAEC Compatibility Program is based on three types of functionality – Content Creation, Content Storage, and Content Consumption – with each targeting a different usage of the MAEC Language. These functionalities are outlined in the table below. Defining the functionalities in this way enables members of the MAEC community to understand how a given Capability is using the MAEC Language and to better determine whether the Capability might suit their needs.

Content Creation	A Tool or Service that creates or aids in the process of creating new MAEC files, including those that consolidate existing MAEC Output Format documents into a single file. The following sub-types of the Content Creation functionality are defined: • **Static Analysis Content Creation:** a Tool or Service that performs some static analysis of one or more input Malware Instances and outputs the results in a MAEC Output Format document. • **Dynamic Analysis Content Creation:** a Tool or Service that performs some dynamic analysis (i.e., instrumented execution) of an input Malware Instance and outputs the results in a MAEC Output Format document. • **Authoring Content Creation:** a Tool or Service that supports the manual creation and editing of MAEC Output Format documents.
Content Storage	A Repository of MAEC Content made available to the community (free or pay).
Content Consumption	A Tool or Service that accepts MAEC Output Format documents as input and either displays their content to the User or uses them to perform some action (remediation, SIM, etc.).

Common Compatibility Requirements

The following requirements apply to all Capabilities that implement support for MAEC, regardless of the specific functionality that is implemented (functionality-specific requirements are given in the "Specific Compatibility Requirements" section below). If a Capability is shown to satisfy all applicable requirements, then the Capability Owner shall receive formal acknowledgement of MAEC compatibility from the Review Authority.

General

These requirements deal with general aspects of MAEC compatibility.

1.1 — The Capability Owner SHALL be a valid legal entity (i.e., an organization or a specific individual, with a valid phone number, email address, and street address).

1.2 — The Capability Owner SHALL agree to adhere to all of the mandatory MAEC compatibility requirements, including the mandatory requirements applicable to the specific functionality.

1.3 — The Capability Owner SHALL provide the Review Authority with a technical point of contact who is qualified to answer questions regarding any MAEC-related functionality of the Capability and coordinate the preparation of the Capability for Correctness Testing.

1.4 — The Capability Owner SHALL provide the Review Authority with a completed "MAEC Compatibility Questionnaire Form." This form will be provided to the Capability Owner after the “MAEC Compatibility Declaration Form” has been processed by the Review Authority.

1.5 — The Capability Owner SHALL work with the Review Authority to make the Capability available for Correctness Testing.

1.6 — The Capability Owner SHALL provide the Review Authority with free access to items needed to perform Correctness Testing, including the Test Results and/or Review Samples, in order to determine compliance with all associated compatibility requirements.

1.7 — As part of receiving formal acknowledgement of MAEC-compatibility, the Capability Owner SHALL agree to support the Review Authority in follow-on testing activities where appropriate types of files will be exchanged with other organizations attempting to prove the correctness of their Capability. This will be managed by the Review Authority and kept to reasonable levels of effort for all involved.

1.8 — The Capability SHALL be available to the public or a set of consumers.

1.9 — The Capability SHALL clearly state the Review Version(s) of MAEC and the associated schema(s) with which it is compatible.

Miscellaneous
These requirements deal with miscellaneous aspects of MAEC compatibility.

2.1 — If the Capability does not satisfy all of the applicable requirements above (1.1 through 1.10), then the Capability Owner SHALL NOT advertise the Capability as MAEC-compatible.

2.2 — If the Capability does not satisfy the requirements specific to its functionality (defined in the sections below), then the Capability Owner SHALL NOT advertise the Capability as MAEC-compatible.

2.3 — The Capability Owner MUST have formal approval from the Review Authority before advertising the Capability as MAEC-compatible.

Correctness
These requirements deal with errors in correctness related to MAEC compatibility, including but not limited to errors relating to schema validation and invalid uses of particular MAEC structures and elements.

3.1 — The Capability Owner SHALL have in place a means for the User to submit correctness errors found in the use of MAEC and in any MAEC content being produced by the Capability.

3.2 — The Capability Owner SHALL have a plan in place to address any correctness errors reported to it.

3.3 — The Capability Owner SHALL address any correctness errors reported to it within a reasonable time frame after the error is initially reported.

Documentation
The following requirements apply to documentation that is provided with a MAEC-compatible Capability.

4.1 — The Capability SHALL include in its documentation a brief description of MAEC and MAEC Compatibility, which can include verbatim portions of documents from the MAEC Web site.

4.2 — The Capability SHALL clearly state in its documentation the extent of its coverage of MAEC and its associated schemas, including those imported from the community efforts of CybOX and MMDEF, either through the elements or individual CybOX objects that it does not support or through the elements and CybOX objects that it does support. For example, if a Capability is applying for formal acknowledgement of MAEC-compatibility as a Dynamic Analysis Content Creation Tool or Service and does not support the CybOX File object and/or the actions associated with the CybOX File object, then the Capability documentation SHALL explicitly state this incompatibility.

4.3 — The Capability SHALL clearly state in its documentation the procedure that a User must follow to submit correctness errors found in any MAEC content being produced by the Capability.

4.4 — If the documentation included with the Capability includes an index, then it SHALL include references to MAEC-related documentation under the term "MAEC."

Validity
The following requirements stem from the requirement that MAEC-compatible Capabilities work with valid documents. Such requirements help to ensure that information is being formatted correctly and that the structure of the document follows the MAEC Language.

5.1 — The Capability SHALL validate all MAEC content (both created and consumed) using W3C XML Schema validation against the version of the MAEC Language with which it is stated to comply.

5.2 — The Capability SHALL report any W3C XML Schema validation errors to the User.

5.3 — The Capability SHALL validate all MAEC content (both created and consumed) using Schematron validation against the version of the MAEC Language with which it is stated to comply.

5.4 — The Capability SHALL report any Schematron validation errors to the User.

Specific Compatibility Requirements

The following requirements only apply to Capabilities for which Capability Owners are seeking MAEC compatibility with respect to the related functionality. A MAEC-compatible Capability MUST provide at least one specific functionality - Content Creation, Content Storage, or Content Consumption.

General Content Creation

These requirements apply to all Tools and Services that intend to create MAEC Content.

6.1 — A Tool or Service that provides MAEC Content MUST generate at least one type of MAEC Output Format (MAEC Bundle, Package, or Container).

6.2 — A Tool or Service that intends to provide technical analysis output for a single Malware Instance and does not intend to capture information on its own attributes SHOULD generate a single MAEC Bundle for the Malware Instance.

6.3 — A Tool or Service that intends to provide output for one or more Malware Instances and/or intends to capture information on its own attributes SHOULD generate one or more MAEC Packages with one or more embedded MAEC Malware Subjects for each Malware Instance that it analyzes. If it does not generate MAEC Packages, then it MUST generate MAEC Containers that contain embedded MAEC Packages.

6.4 — A Tool or Service that intends to provide output for more than one set or group of Malware Instances SHOULD generate one or more MAEC Containers with one or more embedded MAEC Packages for each set or group of Malware Instances that it analyzes.

6.5 — A Tool or Service that intends to capture information on its own attributes MUST document, at a minimum, its name, version, and vendor using the appropriate entities in the MAEC Malware Subject and consequently MUST generate MAEC Packages or Containers of embedded MAEC Packages.

6.6 — A Tool or Service that generates MAEC Packages SHOULD be capable of generating stand-alone MAEC Bundles.

6.7 — A Tool or Service that generates MAEC Containers SHOULD be capable of generating stand-alone MAEC Packages.

6.8 — A Tool or Service SHOULD use its own unique constant namespace portion of the ID across all MAEC Content that it generates.

Static Analysis Content Creation

These requirements apply to all static analysis Tools and Services that intend to create MAEC Content.

7.1 — When generating a MAEC Output Format file, a static analysis Tool or Service SHOULD report its findings using the most appropriate MAEC entities (including but not limited to,

MAEC Actions, Objects, Behaviors, and/or AV Classifications) as well as the most appropriate MAEC Output Format.

Dynamic Analysis Content Creation
These requirements apply to all dynamic analysis Tools and Services that intend to create MAEC content.

8.1 — When generating a MAEC Output Format file, a dynamic analysis Tool or Service SHOULD report its findings using the most appropriate MAEC entities (including but not limited to, MAEC Actions and Behaviors) as well as the most appropriate MAEC Output Format.

Authoring Content Creation
These requirements apply to all Tools and Services that intend to create MAEC Content or help facilitate the creation or modification of MAEC Content.

9.1 — An authoring Tool or Service SHOULD encourage the reuse of existing Malware Subjects, Behaviors, Actions, Objects, and Candidate Indicators.

9.2 — An authoring Tool or Service SHOULD allow the User to invoke validation on a document that is written for the MAEC Language and SHOULD report all W3C XML Schema and Schematron errors to the User.

9.3 — An authoring Tool or Service SHALL allow the User to import and modify existing MAEC content (this includes all MAEC Output Formats).

9.4 — An authoring Tool or Service SHALL allow the User to export the content created as valid MAEC Output Format documents.

9.5 — An authoring Tool or Service SHOULD allow the User to create a document in any MAEC Output Format.

9.6 — An authoring Tool or Service SHOULD report duplicate content to the User.

9.7 — An authoring Tool or Service SHALL provide value and capability above and beyond the capability of a XML editor, as determined by the Review Authority.

Content Storage
These requirements apply to all Repositories that intend to provide a collection of MAEC content.

10.1 — Each MAEC Container, Package, Malware Subject, Analysis, Bundle, Action, Object, Behavior, Candidate Indicator, Behavior Collection, Action Collection, Object Collection, and Candidate Indicator Collection SHALL contain a unique ID with respect to all other MAEC Containers, Packages, Malware Subjects, Analysis, Bundles, Actions, Objects, Behaviors, Candidate Indicators, Behavior Collections, Action Collections, Object Collections, and Candidate Indicator Collections in the Repository.

10.2 — The namespace portion of the ID SHALL be constant across all MAEC content and SHOULD be unique to the Repository.

10.3 — Each MAEC Container, Package, Malware Subject, Analysis, Bundle, Action, Object, Behavior, Candidate Indicator, Behavior Collection, Action Collection, Object Collection, and Candidate Indicator Collection SHALL have the same ID across its existence. An existing item SHOULD NOT be rewritten for some other purpose as Users may be referencing the item in their own content.

10.4 — The Repository owner SHALL document the process by which a User can retrieve content updates.

Content Consumption

These requirements apply to all Tools and Services that intend to consume MAEC content. Note the distinction between "consume" (process information in an intelligent way) and "parse" (extract particular content from a larger document).

11.1 — A Tool or Service that consumes MAEC Content MUST consume at least one type of MAEC Output Format (Bundle, Package, or Container).

11.2 — A Tool or Service that consumes MAEC Content MUST support the parsing of each type of MAEC Output Format to extract any embedded types that it consumes, regardless of the types' location in the Output Format document. For example, a Tool or Service that consumes only Bundles must be able to also parse Packages and Containers to extract Bundle content.

11.3 — If a Tool or Service requires only technical analysis information associated with a Malware Instance, it SHOULD consume MAEC Bundles.

11.4 — If a Tool or Service requires technical analysis information associated with a Malware Instance as well as analysis metadata and relationship information, it SHOULD consume MAEC Packages.

11.5 — If a Tool or Service requires analysis information associated with multiple sets or groups of Malware Instances, it SHOULD consume MAEC Containers.

11.6 — If the Tool or Service does not consume MAEC Output Format files at runtime, the Capability Owner SHALL document the process by which a User can submit MAEC Output Format files to the Capability Owner for interpretation by the Tool or Service. Documentation MUST state how quickly files submitted to the Capability Owner are made available to the Tool or Service.

Review Authority Requirements

The following are requirements pertaining to MAEC compatibility that a Review Authority must adhere to.

12.1 — A Review Authority SHALL clearly identify the Review Version of the capability, and the version of the requirements document that were used to determine formal adherence to the MAEC compatibility requirements for each Capability.

12.2 — The Review Authority SHALL specify the functionality type(s) of the Capability (Content Creation, Content Storage, or Content Consumption).

12.3 — A Review Authority SHALL define and publish sample test materials.

12.4 — A Review Authority SHALL publicize information on how to participate in Correctness Testing so that organizations can prepare as much in advance as possible.

12.5 — A Review Authority SHALL provide a point of contact for arranging Correctness Testing for Capabilities declaring support for MAEC that have completed the "MAEC Compatibility Questionnaire Form."

12.6 — A Review Authority MAY re-test a Capability that has been formally acknowledged of MAEC compatibility at its own discretion.

Revocation

If a Review Authority has approved a Capability as MAEC-compatible, but at a later time the Review Authority has evidence that the requirements are no longer being met, then the Review Authority may revoke its approval and the Capability will no longer be formally acknowledged as MAEC-compatible. The following are the requirements that the Review Authority must follow in order to revoke the acknowledgement.

13.1 — The Review Authority SHALL provide the Capability Owner with a warning of revocation at least two (2) months before revocation is scheduled to occur.

13.2 — The Review Authority MAY delay the date of revocation.

13.3 — If the Review Authority has found that the actions or claims of the Capability Owner are intentionally misleading, then the Review Authority MAY omit the warning period. The Review Authority MAY interpret the phrase "intentionally misleading" as it wishes.

13.4 — If the Review Authority determines that the actions of the Capability Owner with respect to the compatibility requirements are intentionally misleading then revocation SHALL last a minimum of one year.

13.5 — The Review Authority SHALL identify the specific requirements that are not being met.

13.6 — If the Capability Owner believes that the requirements are being met, then the Capability Owner SHALL respond to the warning of revocation by providing specific details that indicate why the Capability meets the requirements under question.

13.7 — If during the warning period, the Capability Owner modifies the Capability such that it complies with the requirements in question, then the Review Authority SHOULD terminate the revocation action for the Capability.

13.8 — The Review Authority SHALL publicize the fact that the formal acknowledgement of MAEC compatibility has been revoked for the Capability.

13.9 — The Review Authority MAY publicize the reason for revocation.

How to Declare Your Product, Service, or Repository "MAEC-Compatible"

To begin the MAEC Compatibility process, send an email to maec@mitre.org requesting the "MAEC Compatibility Declaration Form." This form, along with a copy of the "Requirements and Recommendations for MAEC Compatibility," will be sent to you for review. Once the form has been completed, email it back to maec@mitre.org for processing.

Additional Information

For additional information, please about see the MAEC Compatibility Program and MAEC Technical Use Cases, or contact us at maec@mitre.org.