International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1546

(01/2014)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Cybersecurity information exchange –
Event/incident/heuristics exchange

Malware attribute enumeration and characterization

Recommendation ITU-T X.1546

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	X.1030–X.1049
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1339
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	**X.1540–X.1549**
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1546

Malware attribute enumeration and characterization

Summary

The malware attribute enumeration and characterization (MAEC) language includes enumerations of malware attributes and behaviour that provide a common vocabulary. These enumerations are at different levels of abstraction: low-level observables, mid-level behaviours and high-level taxonomies. Recommendation ITU-T X.1546, which is the initial version of MAEC, focuses on the creation of the enumeration of low-level malware attributes, and leverages the few instances of similar work already done in this area. Thus it will initially be capable of characterizing the most common malware types, including Trojans, worms and rootkits, but it will ultimately be applicable to more esoteric malware types.

History

Edition	Recommendation	Approval	Study Group	Unique ID[*]
1.0	ITU-T X.1546	2014-01-24	17	11.1002/1000/12038

[*] To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Introduction

Recommendation ITU-T X.1546 on the use of malware attribute enumeration and characteristics (MAEC) is an international, information security, community standard to promote open and publicly available security content about malware and malware behaviours. This Recommendation also aims to standardize the transfer of this information across the entire spectrum of security tools and services that can be used to monitor and manage defences against malware. MAEC is a language used to encode malware relevant details.

The MAEC language aims to: 1) improve human-to-human, human-to-tool, tool-to-tool, and tool-to-human communication about malware, 2) reduce potential duplication of malware analysis efforts by researchers, and 3) allow for the faster development of countermeasures by enabling the ability to leverage responses to previously observed malware instances. Threat analysis, intrusion detection, and incident management are processes that deal with all manners of cyberthreats. MAEC, through its uniform encoding of malware attributes, provides a standardized format for the incorporation of actionable information regarding malware in these processes.

Malicious software – also called "malware" – has existed in one form or another since the advent of the first PC virus in 1971. It is presently responsible for a host of malicious activities, ranging from the vast majority of spam email distribution through botnets, to the theft of sensitive information via targeted social engineering attacks. Effectively an autonomous agent operating on behalf of the attacker, malware has the ability to perform any action that is capable of being expressed in code, and as such it represents a prodigious threat to cybersecurity.

The protection of computer systems from malware is therefore currently one of the most important information security concerns for organizations and individuals because even a single instance of uncaught malware can result in damaged systems and compromised data. Being disconnected from a computer network does not completely mitigate this risk of infection, as exemplified by malware that makes use of USB as its insertion vector. As such, the main focus of the majority of anti-malware efforts to date has been on preventing damaging effects through early detection.

There are currently several common methods used for malware detection, based mainly on physical signatures and heuristics. These methods are effective in terms of their narrow scope, although they have their own individual drawbacks, such as the fact that signatures are unsuitable for dealing with zero-day, targeted, polymorphic and other forms of emerging malware. Similarly, heuristic detection may be able to generically detect certain types of malware while missing those that it does not have patterns for, such as kernel-level rootkits. Therefore, it would be safe to say that these methods, while still useful, cannot be exclusively relied upon to deal with the current influx of malware.

Malware attribute enumeration and characterization (MAEC, pronounced "mike") aims to eliminate the ambiguity and inaccuracy that currently exists in malware descriptions and to reduce reliance on signatures. In this way, MAEC seeks to improve human-to-human, human-to-tool, tool-to-tool, and tool-to-human communication about malware; reduce potential duplication of malware analysis efforts by researchers; and allow for the faster development of countermeasures by enabling the ability to leverage responses to previously observed malware instances. As will be illustrated, the MAEC language enables correlation, integration and automation for sharing structured information about malware based upon attributes such as behaviours, artefacts and attack patterns.

As shown in Figure 1, MAEC is composed of a data model that spans several interconnected schemas, thus representing the grammar that defines the language. These schemas permit different forms of MAEC output to be generated, which can be considered as specific uses of the aforementioned grammar.

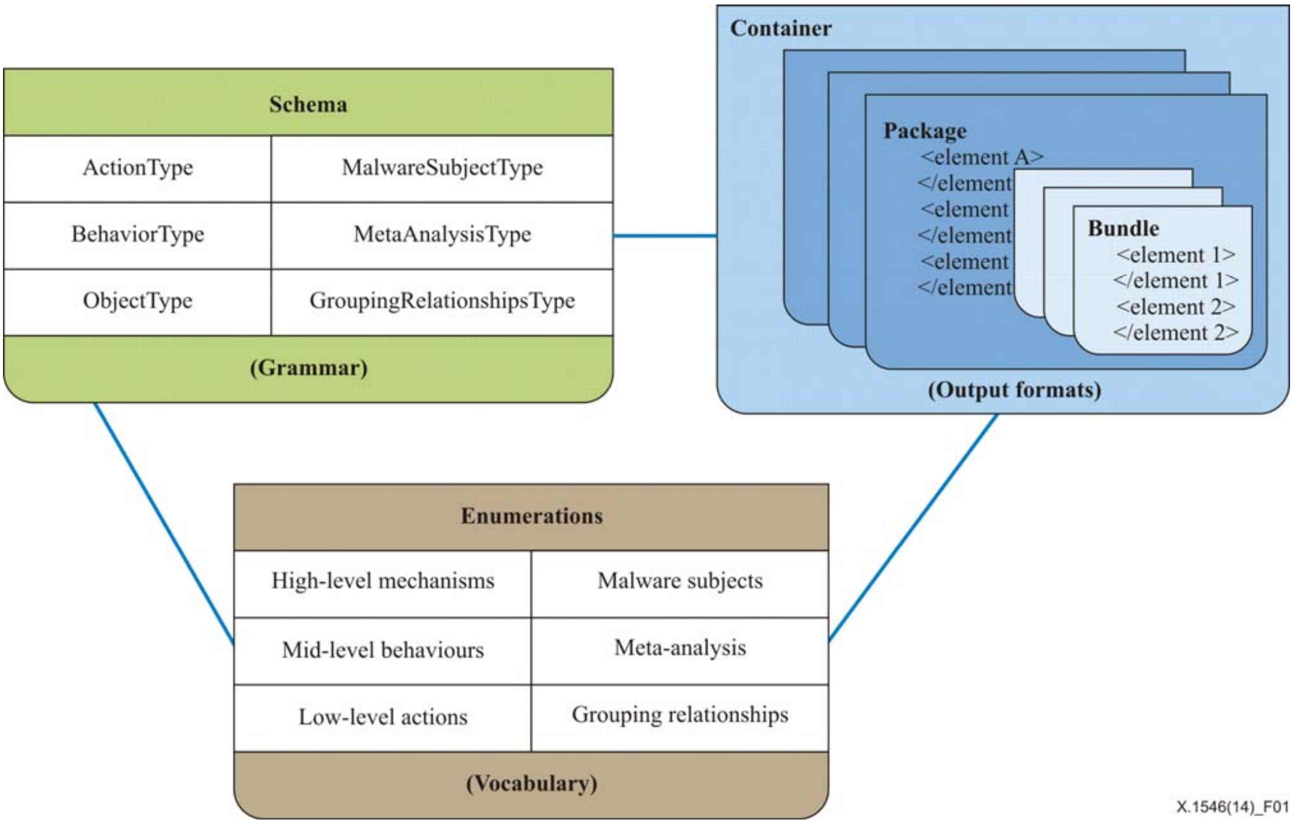

Figure 1 – High-level MAEC overview

The MAEC container, MAEC package and MAEC bundle schemas are targeted at different use cases and thus capture different types of malware-related information.

The MAEC language is related to both the cyber-observable expression (CybOX) language and to the IEEE ICSG's malware metadata exchange format (MMDEF).

CybOX is a standardized language for the specification, capture, characterization and communication of events or stateful properties that are observable in the operational domain. Cyber-observables apply to numerous domains: threat assessment and characterization (detailed attack patterns), malware characterization, operational event management, logging, cyber situational awareness, incident response, digital forensics, and cyberthreat information sharing, among others.

Almost every field in CybOX is optional, so one can use whatever is appropriate and ignore the rest. CybOX can be used to specify and characterize a wide range of cyber objects and can be used to define relational and logical compositions of multiple objects, actions, events, and/or observables.

Malware characterization with MAEC relies on the common mechanism (structure and content) that CybOX provides for addressing cyber-observables across and among MAEC's full range of use cases. Whereas MAEC provides analysis context, indicators, behaviours, and mechanisms, CybOX provides general actions and objects used in the operational cyber domain. A cyber-observable is a *measurable event* or *stateful property* in the cyber domain. Examples of measurable events include registry key creation, file deletion, and the reception of an HTTP GET request; examples of stateful properties include the MD5 hash of a file, the value of a registry key, and the existence of a mutex.

MAEC imports and extends the CybOX object and action. An extremely simplified overview of the CybOX schema is shown in Figure 2; the CybOX components that MAEC uses are shown in green.

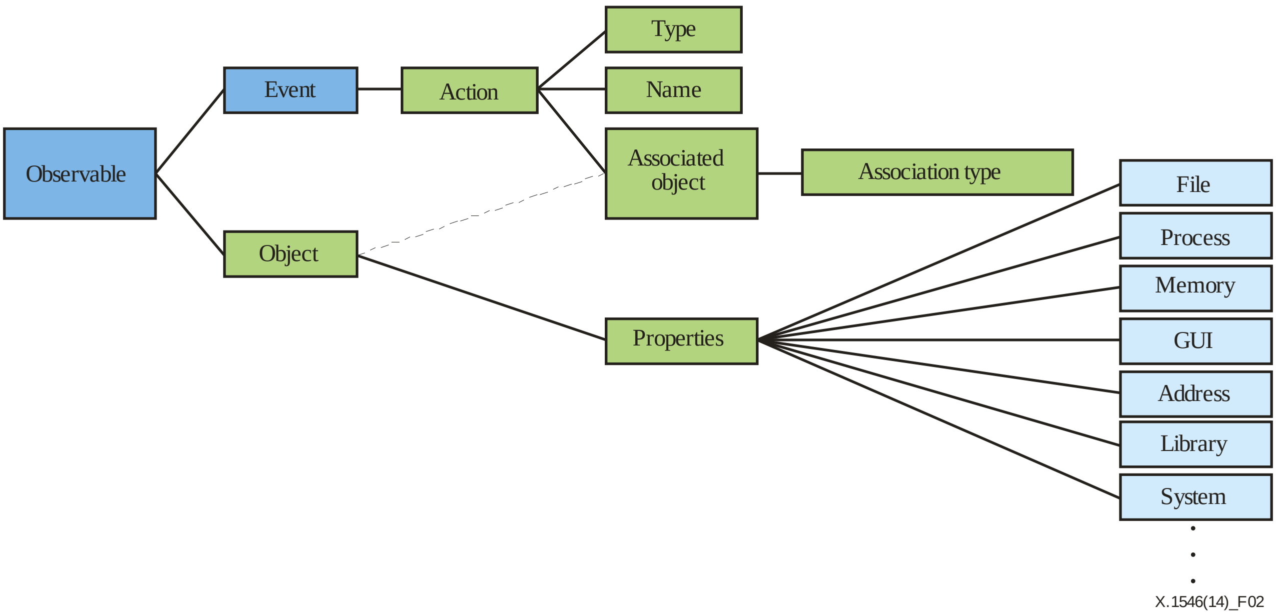

Figure 2 – Cyber-observable expression (CybOX) schema: simple overview

The CybOX properties construct is an abstract placeholder for various predefined Object type schemas (e.g., file, process, memory) that can be instantiated in its place. Properties' schemas, shown in light blue, are maintained independently of the core CybOX schema.

MMDEF is being developed by the Institute of Electrical and Electronics Engineers' (IEEE) Industry Connections Security Group (ICSG). The development of the original schema was led primarily by a group of antivirus (AV) product vendors for the purpose of having some way to augment shared malware samples with additional metadata. As such, it permits the characterization of some static features like hashes and file names, along with some very basic behavioural features.

The information security community contributes to the development of MAEC by participating in the creation of the MAEC language on the MAEC developer's discussion lists and collaboration portal and by integrating the MAEC language into their tools and repository capabilities. The MAEC community includes representatives from a broad spectrum of industry, academia and government organizations from around the world that oversees and collaborates on the MAEC language and the MAEC utilities and tools through the MAEC publicly available repository. This means that MAEC reflects the insights and combined expertise of the broadest possible collection of malware analysis and prevention professionals worldwide.

This Recommendation has been developed on a collaborative basis with The MITRE Corporation bearing in mind the importance of maintaining, to the extent possible, technical compatibility between this Recommendation and the "Requirements and Recommendation for MAEC Compatibility", version 1.1, 7 July 2013

[https://maec.mitre.org/compatible/Requirements_for_MAEC_Compatibility_V1.1.pdf].

Recommendation ITU-T X.1546

Malware attribute enumeration and characterization

1 Scope

This Recommendation provides a structured means to promote open and publicly available security content about malware and malware behaviours, and to standardize the transfer of this information across the entire spectrum of security tools and services that can be used to monitor and manage defences against malware.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ISO/IEC 19757-3]	ISO/IEC 19757-3:2006, *Information technology – Document Schema Definition Languages (DSDL) – Part 3: Rule-based validation – Schematron.*
[W3C XML Schema]	W3C XML Schema Part 2 (2004), *W3C XML Schema Part 2: Datatypes*, Second Edition. <http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 owner [b-ITU-T X.1520]: The custodian (real person or company) having responsibility for the capability.

3.1.2 user [b-ITU-T X.1520]: A consumer or potential consumer of the capability.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 capability: A specific function or functions of a product, service or repository.

3.2.2 capability test results: Data representing the outcome of correctness testing.

3.2.3 content: Any form of a malware attribute enumeration and characteristics (MAEC) entity, including MAEC output format documents, as well as embedded elements/types.

3.2.4 correctness testing: The process of determining whether a tool has correctly implemented MAEC.

3.2.5 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 behaviours or actions, and any related MAEC objects.

3.2.6 MAEC container: A standard form of MAEC output to capture one or more MAEC packages.

3.2.7 MAEC output format: Any of the three standard forms of MAEC output, including the MAEC container, package or bundle.

3.2.8 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.

3.2.9 malware instance: A specific copy of malware.

3.2.10 malware element: A behaviour, attribute, exploit, payload etc., that is related to a specific malware instance, family or class of malware instances.

3.2.11 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.

3.2.12 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.

3.2.13 product: Any anti-malware tool, service or repository that has one or more capabilities.

3.2.14 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 behavioural patterns, the set of malware instances analysed 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.

3.2.15 review: The process of determining whether a capability is MAEC-compatible.

3.2.16 review authority: An entity that performs correctness testing and is authorized to formally acknowledge that a capability is MAEC-compatible.

3.2.17 review sample: A copy of a capability's output provided to a review authority for use in determining whether the capability is MAEC-compatible.

3.2.18 review version: The dated version of MAEC that is being used for determining MAEC compatibility of a capability.

3.2.19 service: A malware analysis, detection or remediation activity that implements one or more capabilities.

3.2.20 tool: A software application or device that implements one or more functionalities. A tool analyses, detects or remediates malware through various methods, e.g., a static analysis tool, dynamic analysis tool, signature based scanner, heuristic based scanner, etc. A tool can also perform content authoring.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

AV	Antivirus
CybOX	Cyber-Observable expression
ID	Identifier
MAEC	Malware Attribute Enumeration and Characterization
MMDEF	Malware Metadata Exchange Format

SIM	Security Information Management
XML	extensible Markup Language

5 Conventions

MAEC is used as a noun in this Recommendation.

6 High-level requirements

The following items define the concepts, roles and responsibilities related to five different capabilities, each targeting a different use of the MAEC language, that comprise the proper use of the MAEC language. These capabilities enable members of the MAEC community to easily understand how a given product is using the MAEC language and how it might suit their needs.

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 below in clause 10, "Specific compatibility requirements"). 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.

Prerequisites

6.1 The capability owner shall be a valid legal entity, i.e., an organization or a specific individual, with a valid phone number, e-mail address and street mail address.

6.2 The capability owner shall agree to adhere to all of the mandatory MAEC compatibility requirements, including the mandatory requirements applicable to a specific functionality.

6.3 The capability owner shall provide the review authority with a technical point of contact that is qualified to answer questions regarding any MAEC-related functionality of the capability and to coordinate the preparation of the capability for correctness testing.

6.4 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.

6.5 The capability owner shall work with the review authority to make the product, service or repository available for correctness testing.

6.6 The capability owner shall provide the review authority with free access to items needed to perform correctness testing, including the test results and/or the review samples, in order to determine compliance with all associated compatibility requirements.

6.7 As part of receiving formal acknowledgement of MAEC-compatibility, the capability owner shall agree to support the review authority in follow-up 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.

6.8 The capability shall be available to the public or a set of consumers.

6.9 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.

6.10 If the capability does not satisfy all of the applicable requirements above (6.1 to 6.9), then the capability owner shall not advertise the capability as MAEC-compatible.

6.11 If the capability does not satisfy the requirements specific to its functionality (defined in clauses 10.1 to 10.27), then the capability owner shall not advertise the capability as MAEC-compatible.

6.12 The capability owner shall have formal approval from the review authority before advertising the capability as MAEC-compatible.

7 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.

7.1 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.

7.2 The capability owner shall have a plan in place to address any correctness errors reported to it.

7.3 The capability owner shall address any correctness errors reported to it within a reasonable time frame after the error was initially reported.

8 Documentation

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

8.1 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 website.

8.2 The capability shall clearly state in its documentation its coverage of MAEC and its associated schemas, including those imported from the community efforts of cyber-observable expression (CybOX) and malware metadata exchange format (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 it 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.

8.3 The capability shall clearly state in its documentation the procedure that a user should follow to submit correctness errors found in any MAEC content being produced by the product.

8.4 If the documentation included with the capability includes an index, then it shall include references to MAEC-related documentation under the term "MAEC."

9 Validity

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

9.1 The capability shall validate all MAEC content (both produced and consumed) using W3C XML schema validation (see [W3C XML Schema]) against the version of the MAEC language with which it is stated to comply.

9.2 The capability shall report any W3C XML schema validation errors to the user.

9.3 The capability shall validate all MAEC content (both created and consumed) using Schematron validation (see [ISO/IEC 19757-3]) against the version of the MAEC language with which it is stated to comply.

9.4 The capability shall report any Schematron validation errors to the user.

10 Specific capability 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 shall provide at least one specific functionality: content creation, content storage or content consumption.

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 subtypes 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 paying).
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, security information management (SIM), etc.).

General content creation

These requirements apply to all tools and services that intend to provide MAEC content output.

10.1 A tool or service that provides MAEC content shall generate at least one type of MAEC output format (MAEC bundle, package or container).

10.2 Each tool or service that intends to provide 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.

10.3 A tool or service that intends to provide output for a single malware instance 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 analyses. If it does not generate MAEC packages, then it shall generate MAEC containers that contain embedded MAEC packages.

10.4 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 analyses.

10.5 A tool or service that intends to capture information on its own attributes shall document, at a minimum, its name, version and vendor using the appropriate entities in the MAEC malware subject and consequently shall generate MAEC packages or containers of embedded MAEC packages.

10.6 A tool or service that generates MAEC packages should be capable of generating stand-alone MAEC bundles.

10.7 A tool or service that generates MAEC containers should be capable of generating stand-alone MAEC packages.

10.8 A tool or service should use its own unique constant namespace portion of the identifier (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.

10.9 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, behaviours, 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.

10.10 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 behaviours), 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.

10.11 An authoring tool or service should encourage the reuse of existing malware subjects, behaviours, actions, objects and candidate indicators.

10.12 An authoring tool or service should allow the user to invoke validation on a document that is written for the MAEC language and report all W3C XML schema and Schematron errors to the user.

10.13 An authoring tool or service shall allow the user to import and edit existing MAEC content (this includes all MAEC output formats).

10.14 An authoring tool or service shall allow the user to export the content created as valid MAEC output format documents.

10.15 An authoring tool or service should report duplicate content to the user.

10.16 An authoring tool or service shall provide value and capability above and beyond the capability of an extensible markup language (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.

10.17 Each MAEC container, package, malware subject, analysis, bundle, action, object, behaviour, candidate indicator, behaviour 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, behaviours, and candidate indicators, behaviour collections, action collections, object collections and candidate indicator collections in the repository.

10.18 Each MAEC action and object should contain an ID with respect to all other MAEC actions and objects where such an ID is unique with respect to all other MAEC actions and objects in the repository.

10.19 The namespace portion of ID shall be constant across all MAEC content and should be unique to the repository.

10.20 Each MAEC container, package, malware subject, analysis, bundle, action, object, behaviour, and candidate indicators, behaviour 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.

10.21 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).

10.22 A tool or service that consumes MAEC content shall consume at least one type of MAEC output format (bundle, package or container).

10.23 A tool or service that consumes MAEC content shall 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.

10.24 If a tool or service requires only technical analysis information associated with malware instance, it should consume MAEC bundles.

10.25 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.

10.26 If a tool or service requires analysis information associated with multiple sets or groups of malware instances, it should consume MAEC containers.

10.27 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 shall state how quickly files submitted to the capability owner are made available to the tool or service.

11 Review authority requirements

The following are requirements pertaining to MAEC compatibility that a review authority shall adhere to.

11.1 A review authority shall clearly identify the review version of the capability and the version of the MAEC compatibility requirements document, along with the version of the MAEC language that was used to determine formal adherence to the MAEC compatibility requirements for each capability.

11.2 The review authority shall specify the functionality type(s) of the capability (content creation, content storage or content consumption).

11.3 A review authority shall define and publish sample test materials.

11.4 The review authority shall publicize information on how to participate in correctness testing so that organizations can prepare as much in advance as possible.

11.5 The 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."

11.6 The review authority may retest a capability that has been formally acknowledged for its MAEC-compatibility, at its own discretion.

12 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 shall follow in order to revoke the acknowledgement.

12.1 The review authority shall provide the capability owner with a warning of revocation at least two (2) months before revocation is scheduled to occur.

12.2 The review authority may delay the date of revocation.

12.3 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.

12.4 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.

12.5 The review authority shall identify the specific requirements that are not being met.

12.6 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.

12.7 If during the warning period the capability owner modifies the capability so that it complies with the requirements in question, during the warning period, then the review authority should terminate the revocation action for the capability.

12.8 The review authority shall publicize that the formal acknowledgement of the correct compatibility of MAEC has been revoked for the capability.

12.9 The review authority may publicize the reason for revocation.

Bibliography

[b-ITU-T X.1520] Recommendation ITU-T X.1520 (2011), *Common vulnerabilities and exposures*.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Terminals and subjective and objective assessment methods
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2014