Characterizing Malware with MAEC and STIX

v1.0
April 21st, 2014

Introduction

This document describes the use of the Malware Attribute Enumeration and Characterization (MAEC) and Structured Threat Information eXpression (STIX) and languages in the context of malware characterization and malware metadata exchange. By describing the relationships between the languages and by providing details on each language's ability to capture malware-related information, this document answers the question, "When should I use MAEC, when should I use STIX, and when should I use both?"

Document Outline

We begin by providing an executive summary of the usage of STIX, MAEC, and MAEC embedded in STIX in terms of capturing malware-related information. Following this, we provide a high-level overview of the general capabilities, context, and targeted audience of each language (or combination thereof) in this context. Also included in this section is a simple flow chart to help readers determine which language best fits their needs. The next section provides more details on each language in this context, including specific details of how they should be used. Finally, we discuss the relationship between each of the languages, including their mutual use of CybOX.

Executive Summary

MAEC and STIX were designed with very dissimilar use cases in mind, and thus serve different roles when it comes to capturing information about malware. MAEC is intended to provide a comprehensive, structured way of capturing detailed information about malware samples, and is therefore targeted primarily towards malware analysts. STIX, meanwhile, is meant to capture a broad spectrum of cyber-threat related information, including basic information on malware, which makes it applicable to a more diverse audience.

MAEC content can also be embedded inside of STIX, which permits the two languages to complement each other. When used together in this fashion, they permit the capture of detailed malware information alongside related cyber threat information. This allows for useful, finer-grained relationships between malware and the larger cyber threat context to be established and expressed.

CybOX, on the other hand, provides the common foundation in MAEC and STIX for capturing the observables relevant to each language. While CybOX lacks the ability as a standalone language to capture meaningful context about malware, its usage in both MAEC and STIX enables the interoperability of both languages around malware-related observables.

Options for Capturing Malware Information

Each of STIX and MAEC can be used individually to capture information about malware. However, in some situations, it may be preferable to embed MAEC content within a STIX document. The types of malware-related information captured by each of these options – MAEC, STIX, and MAEC embedded in STIX – are shown in Table 1 below. The table also shows the context provided in each case, as well as the targeted audience.

MAEC™	STIX™	STIX™ + MAEC™
Captures structured, detailed malware information: • Capabilities • Behaviors • Actions • AV Classifications • Extracted Objects • Relationships • Associated Metadata	**Captures unstructured, basic malware information:** • Type • Name • Description	**Captures broad spectrum of malware information:** • Basic, descriptive information via STIX ○ Provides Identification • Detailed, structured information via MAEC ○ Provides broader understanding • E.g., a brief description of a malware family and detailed descriptions of several of its members
Provides analytical context • "What" does the malware do? • "How" does the malware operate?	**Provides surrounding context** • "Who" used the malware? • "Where" was the malware used?	**Provides surrounding AND analytical context** • Connects detailed malware information to broader threat context • E.g., "what" specific features of a malware instance are associated with a particular threat actor?
Target audience: • Malware Analysts/Reverse Engineers	**Target audience:** • Cyber Threat/Intelligence Analysts • SOC/CERT Operators • Incident Responders	**Target audience:** • Malware Analysts/Reverse Engineers • Cyber Threat/Intelligence Analysts • SOC/CERT Operators • Incident Responders

Table 1: Options for capturing malware information

The flow chart depicted in Figure 2 below is provided to further help readers understand the basic circumstances under which they should use MAEC, STIX, or MAEC embedded in STIX. More details on the use of each option are provided in the sections that follow.

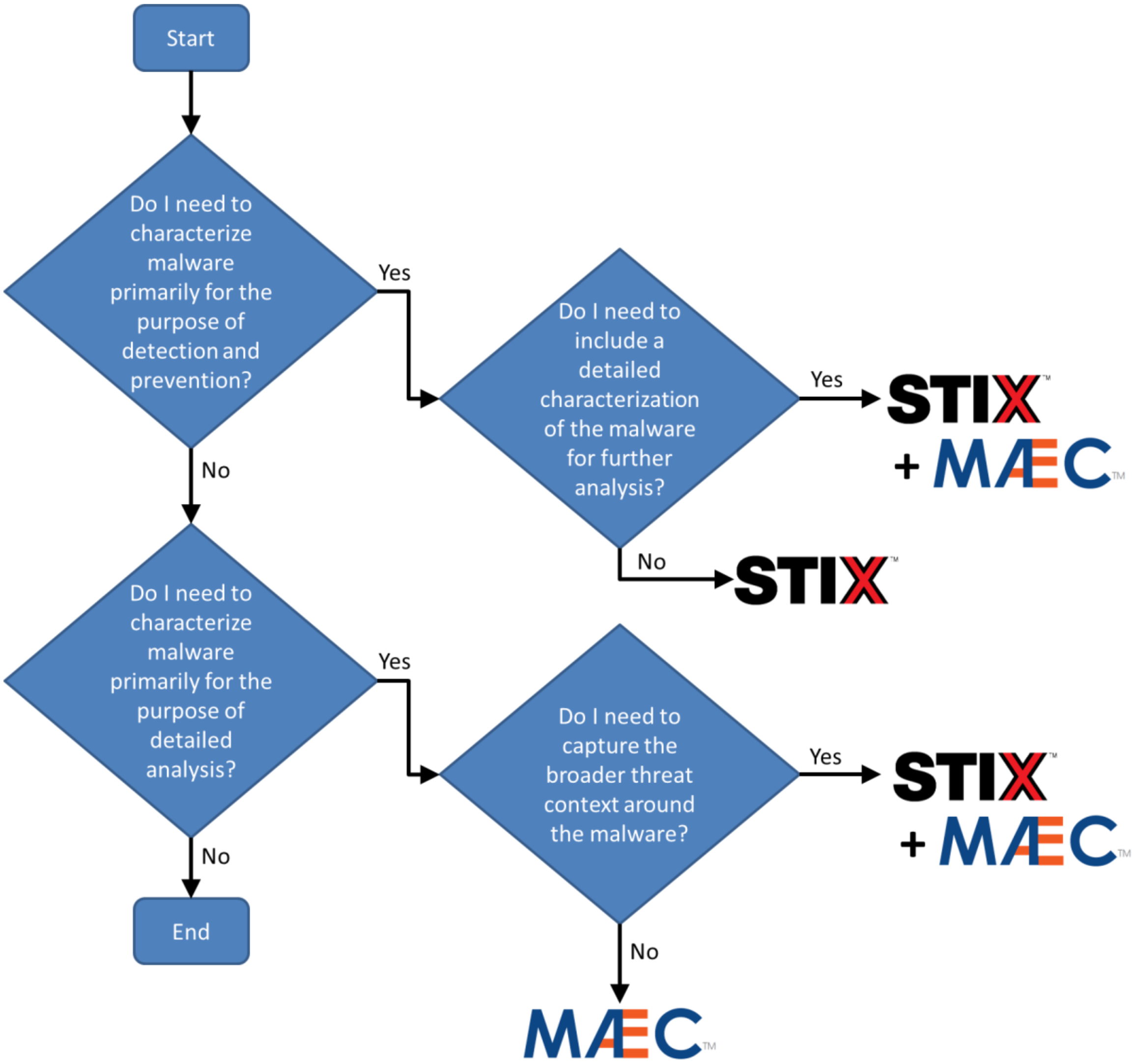

Start
Do I need to characterize malware primarily for the purpose of detection and prevention?
Yes
Do I need to include a detailed characterization of the malware for further analysis?
Yes
STIX + MAEC
No
STIX
No
Do I need to characterize malware primarily for the purpose of detailed analysis?
Yes
Do I need to capture the broader threat context around the malware?
Yes
STIX + MAEC
No
MAEC
No
End

Figure 2: STIX, MAEC, and embedded MAEC in STIX usage flowchart

Using MAEC Individually

MAEC captures malware features, including details of how it operates, in a structured fashion. Typically, MAEC is used to capture the results of various forms of malware analysis, such as the execution of a malware binary in a sandbox or a manual static analysis done by an analyst. In addition, MAEC can also be used to capture secondary information such as prevalence data related to the malware and details of the analysis tools. Thus, MAEC is intended to convey detailed analytical information about malware that can further drive analysis or understanding by either machines or human analysts.

In particular, MAEC should be used in a standalone capacity for the capture and exchange of:

- Detailed information on one or more malware samples (derived from one or more analyses)
  - Static analysis results
  - Dynamic analysis results
  - Manual (i.e., human) analysis results, which may provide higher-level information on the behaviors or capabilities of the malware
- Metadata relating to malware analysis
  - Information on the tools used in the analysis
  - Information on the analysts/organization who performed the analysis
  - Comments and other observations recorded during malware analysis
- Relationships between multiple malware samples
  - Grouping relationships for characterizing clusters of related malware samples

On the other hand, MAEC is *not* intended to capture cyber threat intelligence data associated with malware. Indicators, threat actors, and other such entities that may be tied to malware fall under the domain of STIX. In particular, it is important to understand that a MAEC document, which typically captures raw static and dynamic malware analysis data, is *not* suitable for direct consumption as a malware indicator. While it is true that MAEC can capture the data that may form the basis of a malware indicator, such information almost always needs to be pruned and vetted by a human analyst before it can be used effectively.

MAEC is capable of capturing a much more expansive variety of data than we have highlighted here; please see the MAEC Language Specification and Detailed Examples document for details.

Using STIX Individually

STIX captures an extensive set of information on cyber threats in a standardized and structured manner, including details of indicators, campaigns, threat actors, and TTPs (Tactics, Techniques, and Procedures). STIX can also be used to provide basic identifying information about malware samples, families, or classes so that higher-level cyber threat information can be directly associated with explicit malware samples (for example, a STIX indicator can be associated with the particular malware samples that it is intended to detect). However, standalone STIX content is *not* designed to include a detailed characterization of the malware itself; it is only appropriate for STIX in this fashion to capture more broadly applicable data, such as the threat actors who have been known to use the malware.

In terms of malware-related information, STIX can be used in a standalone capacity for the capture and exchange of:

- TTPs that provide a lightweight description of one or more malware samples
- Indicators pertaining to one or more malware samples, e.g., describing a particular file that is dropped by a malware sample
- Incidents where one or more malware samples were used
- Campaigns associated with one or more malware samples
- Threat actors that made use of one or more malware samples

STIX is able to capture the type (e.g., instance or family) of malware being described, the name of the malware, and a brief description of the malware via the TTP component schema. Generally, a single STIX Package should be created and populated with multiple TTPs – one for each malware instance, family, or class identified. Other high-level STIX entities, such as Indicators, can then reference these malware-related TTPs as needed to provide the basic context of the associated malware entity. Please see the STIX Idioms documentation for examples and further details.

Using MAEC Embedded in STIX

By embedding native MAEC data in a STIX document, detailed, structured information about one or more malware samples can be captured alongside broader cyber threat information. As shown in Table 2 below, there are generally two distinct use cases that are applicable in this context.

	Use Case 1	**Use Case 2**
Primary content	Detailed understanding of one or more malware samples	Context around one or more cyber threat entities
Secondary content	Cyber threat context around malware	Detailed understanding of the malware samples
Target audience	• Malware analysts • Intelligence analysts	• Malware analysts • Intelligence analysts • SOC/CERT operators
Example	Details of a malware sample and the threat actors to which it is attributed	Complete description of several cyber campaigns, including the malware that was used

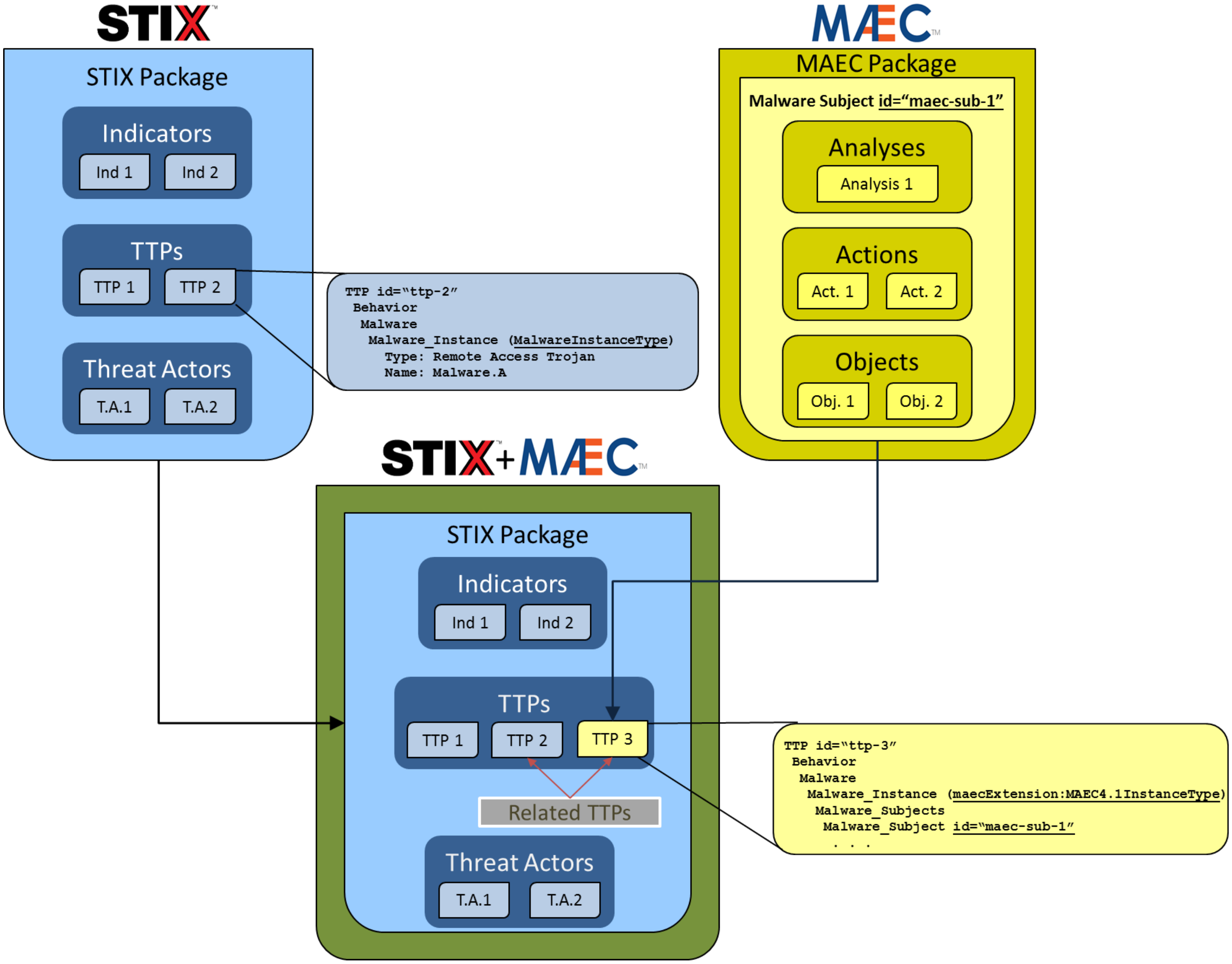
Table 2: Use cases for embedding MAEC data in a STIX document

Generally, STIX together with embedded MAEC data can be used for the capture and exchange of:

- TTPs that capture both a lightweight and detailed description of one or more malware samples
  - The lightweight description is for consumption by intelligence analysts and SOC operators
  - The detailed description is for consumption by malware analysts and reverse engineers
  - Both descriptions can be related to each other for easy future access and correlation
- Relationships between cyber threat entities and the detailed descriptions of any associated malware samples
  - For example, this might include cyber incidents which involved one or more malware samples, along with structured descriptions of the malware samples
- Leads used in malware-based attribution, in conjunction with information about the adversaries themselves
  - MAEC (via its incorporation of CybOX) can capture many low-level details that can provide clues to attribution (e.g., the directory in which the code was compiled)
  - STIX (via its Threat Actor entity) can provide the contextual information about the "who"

When MAEC content is embedded in a STIX document, detailed malware information can still be captured through the TTP component schema. However, instead of using the "MalwareInstanceType" type defined in the TTP schema, one should use the "MAEC4.1InstanceType" type (an extension of the "MalwareInstanceType" type) from the MAEC Malware extension schema. This allows a MAEC Package to be embedded in a STIX TTP for a complete, structured characterization of one or more malware samples (it is recommended that a single MAEC Package be constructed with one Malware Subject defined for each malware sample being characterized).

As depicted in Figure 3 below, in addition to creating a TTP with embedded MAEC data, one may also want to create a corresponding TTP with a simple description of the malware using the existing "MalwareInstanceType" type. The pair of TTPs, one with MAEC content and one a simple description, can then be associated using the STIX "Related_TTPs" field. This provides flexibility in that one may use or exchange either the detailed or the simple TTP while still being able to access the other.



```
TTP id="ttp-2"
 Behavior
  Malware
   Malware_Instance (MalwareInstanceType)
     Type: Remote Access Trojan
     Name: Malware.A
```

```
TTP id="ttp-3"
 Behavior
  Malware
   Malware_Instance (maecExtension:MAEC4.1InstanceType)
    Malware_Subjects
     Malware_Subject id="maec-sub-1"
       . . .
```

Figure 3: Example Usage of MAEC embedded in STIX

Relationships between Languages

The conceptual relationships between STIX, MAEC, and CybOX are illustrated in Figure 4, below. As shown, STIX encapsulates detailed malware characterization data into a broader threat information context through the incorporation of MAEC. Furthermore, MAEC and STIX both use CybOX to represent their cyber observables. For example, MAEC uses CybOX to describe the various objects on which a malware instance may operate, such as files, Windows registry keys, etc. On the other hand, STIX uses CybOX as a standardized way of expressing the objects and patterns that define a particular cyber threat indicator, such as a particular user agent string in malicious HTTP traffic.

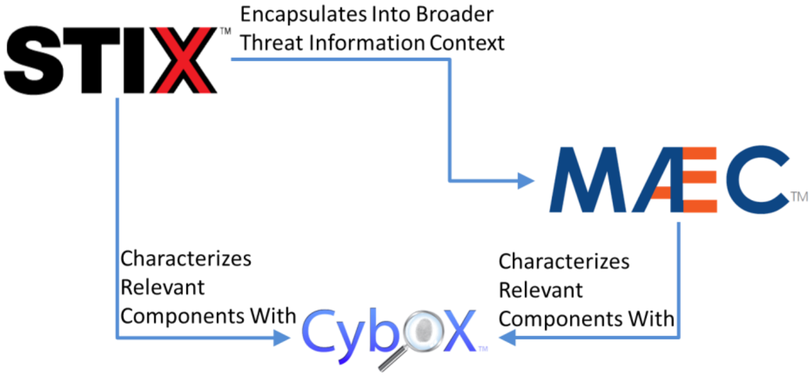

STIX
Encapsulates Into Broader Threat Information Context
MÆC
Characterizes Relevant Components With
Characterizes Relevant Components With
CybOX

Figure 4: Relationships between MAEC, STIX, and CybOX

The mutual use of CybOX by both STIX and MAEC greatly facilitates the construction of malware indicators. As graphically depicted in Figure 5 below, let's say an organization receives a malware sample, which is analyzed with MAEC-enabled tools. Based on the results, an analyst decides that a particular file and registry key created by the malware sample would serve as good indicators. To share these indicators with the organization's partners, the analyst simply captures the CybOX objects from the tool-generated MAEC document as STIX Indicator elements in a STIX document. The common use of CybOX between both languages eliminates the need for any conversion, translation, or other post-processing of the analysis data, thus streamlining the malware analysis to indicator definition process.

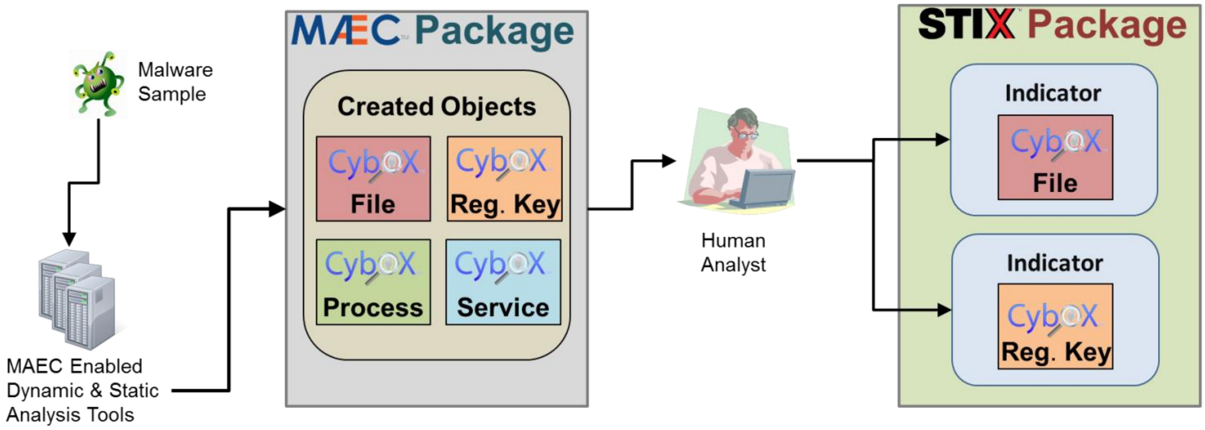


Figure 5: Example MAEC to STIX Indicator workflow