STIX[TM] Version 1.2.1. Part 3: Core

Committee Specification Draft 01

06 November 2015

Specification URIs

This version:

http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part3-core/stix-v1.2.1-csd01-part3-core.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part3-core/stix-v1.2.1-csd01-part3-core.html
http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part3-core/stix-v1.2.1-csd01-part3-core.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part3-core.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part3-core.html
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part3-core.pdf

Technical Committee:

OASIS Cyber Threat Intelligence (CTI) TC

Chair:

Richard Struse (Richard.Struse@HQ.DHS.GOV), DHS Office of Cybersecurity and Communications (CS&C)

Editors:

Sean Barnum (sbarnum@mitre.org), MITRE Corporation
Desiree Beck (dbeck@mitre.org), MITRE Corporation
Aharon Chernin (achernin@soltra.com), Soltra
Rich Piazza (rpiazza@mitre.org), MITRE Corporation

Additional artifacts:

This prose specification is one component of a Work Product that also includes:

- *STIX Version 1.2.1. Part 1: Overview*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part1-overview/stix-v1.2.1-csd01-part1-overview.html
- *STIX Version 1.2.1. Part 2: Common*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part2-common/stix-v1.2.1-csd01-part2-common.html
- *STIX Version 1.2.1. Part 3: Core* (this document). http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part3-core/stix-v1.2.1-csd01-part3-core.html
- *STIX Version 1.2.1. Part 4: Indicator*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part4-indicator/stix-v1.2.1-csd01-part4-indicator.html
- *STIX Version 1.2.1. Part 5: TTP*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part5-ttp/stix-v1.2.1-csd01-part5-ttp.html
- *STIX Version 1.2.1. Part 6: Incident*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part6-incident/stix-v1.2.1-csd01-part6-incident.html
- *STIX Version 1.2.1. Part 7: Threat Actor*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part7-threat-actor/stix-v1.2.1-csd01-part7-threat-actor.html
- *STIX Version 1.2.1. Part 8: Campaign*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part8-campaign/stix-v1.2.1-csd01-part8-campaign.html

- *STIX Version 1.2.1. Part 9: Course of Action*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part9-coa/stix-v1.2.1-csd01-part9-coa.html
- *STIX Version 1.2.1. Part 10: Exploit Target*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part10-exploit-target/stix-v1.2.1-csd01-part10-exploit-target.html
- *STIX Version 1.2.1. Part 11: Report*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part11-report/stix-v1.2.1-csd01-part11-report.html
- *STIX Version 1.2.1. Part 12: Default Extensions*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part12-extensions/stix-v1.2.1-csd01-part12-extensions.html
- *STIX Version 1.2.1. Part 13: Data Marking*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part13-data-marking/stix-v1.2.1-csd01-part13-data-marking.html
- *STIX Version 1.2.1. Part 14: Vocabularies*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part14-vocabularies/stix-v1.2.1-csd01-part14-vocabularies.html
- *STIX Version 1.2.1. Part 15: UML Model*. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part15-uml-model/stix-v1.2.1-csd01-part15-uml-model.html
- UML Model Serialization: http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/uml-model/

Related work:

This specification replaces or supersedes:

- *STIX^{TM} 1.2 Core Specification (v1.2).* https://github.com/STIXProject/specifications/blob/version1.2/documents/pdf%20versions/STIX_Core_Draft.pdf

This specification is related to:

- *CybOX$^{[TM]}$ Version 2.1.1.* Work in progress. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti-cybox
- *CybOX$^{[TM]}$ 2.1.* https://cyboxproject.github.io/

Abstract:

The Structured Threat Information Expression (STIX) framework defines nine core constructs and the relationships between them for the purposes of modeling cyber threat information and enabling cyber threat information analysis and sharing. This specification document defines the Core data model, which defines the STIX Package, the root object for all STIX content.

Status:

This document was last revised or approved by the OASIS Cyber Threat Intelligence (CTI) TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document. Any other numbered Versions and other technical work produced by the Technical Committee (TC) are listed at https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti#technical.

TC members should send comments on this specification to the TC’s email list. Others should send comments to the TC’s public comment list, after subscribing to it by following the instructions at the “Send A Comment” button on the TC’s web page at https://www.oasis-open.org/committees/cti/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC’s web page (https://www.oasis-open.org/committees/cti/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[STIX-v1.2.1-Core]

STIX$^{[TM]}$ Version 1.2.1. Part 3: Core. Edited by Sean Barnum, Desiree Beck, Aharon Chernin, and Rich Piazza. 06 November 2015. OASIS Committee Specification Draft 01. http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part3-core/stix-v1.2.1-csd01-part3-core.html. Latest version: http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part3-core.html.

Notices

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see https://www.oasis-open.org/policies-guidelines/trademark for above guidance.

STIX[™], TAXII[™], AND CybOX[™] (STANDARD OR STANDARDS) AND THEIR COMPONENT PARTS ARE PROVIDED “AS IS” WITHOUT ANY WARRANTY OF ANY KIND, EITHER EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY THAT THESE STANDARDS OR ANY OF THEIR COMPONENT PARTS WILL CONFORM TO SPECIFICATIONS, ANY

IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR FREEDOM FROM INFRINGEMENT, ANY WARRANTY THAT THE STANDARDS OR THEIR COMPONENT PARTS WILL BE ERROR FREE, OR ANY WARRANTY THAT THE DOCUMENTATION, IF PROVIDED, WILL CONFORM TO THE STANDARDS OR THEIR COMPONENT PARTS. IN NO EVENT SHALL THE UNITED STATES GOVERNMENT OR ITS CONTRACTORS OR SUBCONTRACTORS BE LIABLE FOR ANY DAMAGES, INCLUDING, BUT NOT LIMITED TO, DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, ARISING OUT OF, RESULTING FROM, OR IN ANY WAY CONNECTED WITH THESE STANDARDS OR THEIR COMPONENT PARTS OR ANY PROVIDED DOCUMENTATION, WHETHER OR NOT BASED UPON WARRANTY, CONTRACT, TORT, OR OTHERWISE, WHETHER OR NOT INJURY WAS SUSTAINED BY PERSONS OR PROPERTY OR OTHERWISE, AND WHETHER OR NOT LOSS WAS SUSTAINED FROM, OR AROSE OUT OF THE RESULTS OF, OR USE OF, THE STANDARDS, THEIR COMPONENT PARTS, AND ANY PROVIDED DOCUMENTATION. THE UNITED STATES GOVERNMENT DISCLAIMS ALL WARRANTIES AND LIABILITIES REGARDING THE STANDARDS OR THEIR COMPONENT PARTS ATTRIBUTABLE TO ANY THIRD PARTY, IF PRESENT IN THE STANDARDS OR THEIR COMPONENT PARTS AND DISTRIBUTES IT OR THEM “AS IS.”

Table of Contents

1 Introduction

[All text is normative unless otherwise labeled]

The Structured Threat Information Expression (STIX[TM]) framework defines nine top-level component data models: Observable[1], Indicator, Incident, TTP, ExploitTarget, CourseOfAction, Campaign, ThreatActor, and Report. In addition, it defines a core data model for packaging and conveying content from any of these top-level components. This document serves as the specification for the STIX Core data model, which is the unifying data model for all STIX content.

The STIX Core data model defines the concept of a STIX Package, the top-level object that is used to aggregate and convey all other objects of the STIX data models. The STIX Package has two main parts: a set of instances of any of the nine top-level components, which is the content of the STIX Package, and a STIX header, which can provide context for that content.

In Section **1.1** we discuss additional specification documents, in Section **1.2** we provide document conventions, and in Section **1.3** we provide terminology. References are given in Sections **1.4** and **1.5**. In Section **2**, we give background information to help the reader better understand the specification details that are provided later in the document. We present the Core data model specification details in Section **3** and conformance information in Section **4**.

1.1 STIX[TM] Specification Documents

The STIX specification consists of a formal UML model and a set of textual specification documents that explain the UML model. Specification documents have been written for each of the key individual data models that compose the full STIX UML model.

The *STIX Version 1.2.1 Part 1: Overview* document provides a comprehensive overview of the full set of STIX data models, which in addition to the nine data models mentioned in the Introduction, includes a core data model, a common data model, a cross-cutting data marking data model, various extension data models, and a set of default controlled vocabularies. *STIX Version 1.2.1 Part 1: Overview* also summarizes the relationship of STIX to other languages and outlines general STIX data model conventions.

Figure 1-1 illustrates the set of specification documents that are available. The color black is used to indicate the specification overview document, altered shading differentiates the overarching Core and Common data models from the supporting data models (vocabularies, data marking, and default extensions), and the color white indicates the component data models. The solid grey color denotes the overall STIX Language UML model. This STIX Core specification document is highlighted in its associated color (see Section **1.2.3.3**). For a list of all STIX documents and related information sources, please see *STIX Version 1.2.1 Part 1: Overview*.

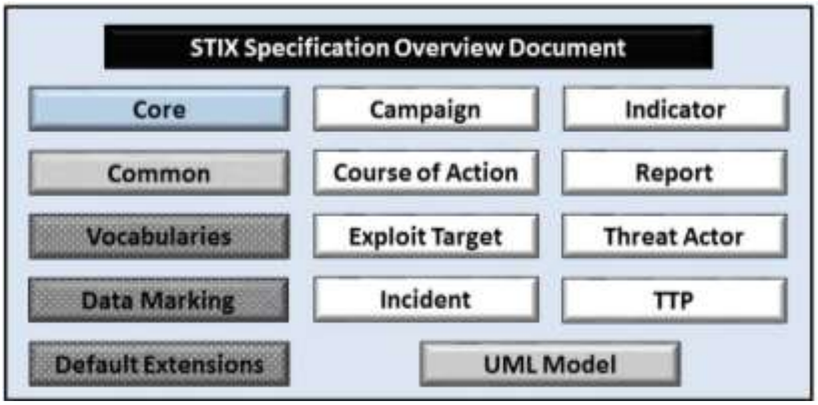


Figure 1-1. STIX[TM] Language v1.2.1 specification documents

1.2 Document Conventions

The following conventions are used in this document.

1.2.1 Fonts

The following font and font style conventions are used in the document:

- Capitalization is used for STIX high level concepts, which are defined in *STIX Version 1.2.1 Part 1: Overview*.

  Examples: Indicator, Course of Action, Threat Actor

- The `Courier New` font is used for writing UML objects.

  Examples: `RelatedIndicatorsType`, `stixCommon:StatementType`

  Note that all high level concepts have a corresponding UML object. For example, the Course of Action high level concept is associated with a UML class named, `CourseOfActionType`.

- The '*italic'* font (with single quotes) is used for noting actual, explicit values for STIX Language properties. The *italic* font (without quotes) is used for noting example values.

  Example: *'PackageIntentVocab-1.0,' high, medium, low*

1.2.2 UML Package References

Each STIX data model is captured in a different UML package (e.g., Core package, Campaign package, etc.). To refer to a particular class of a specific package, we use the format `package_prefix:class`, where `package_prefix` corresponds to the appropriate UML package. *STIX Version 1.2.1 Part 1: Overview* contains a list of the packages used by the Core data model, along with the associated prefix notations, descriptions, examples.

Note that in this specification document, we do not explicitly specify the package prefix for any classes that originate from the Core data model.

1.2.3 UML Diagrams

This specification makes use of UML diagrams to visually depict relationships between STIX Language constructs. Note that the diagrams have been extracted directly from the full UML model for STIX; they have not been constructed purely for inclusion in the specification documents. Typically, diagrams are included for the primary class of a data model, and for any other class where the visualization of its relationships between other classes would be useful. This implies that there will be very few diagrams for classes whose only properties are either a data type or a class from the STIX Common data model. Other diagrams that are included correspond to classes that specialize a superclass and abstract or generalized classes that are extended by one or more subclasses.

In UML diagrams, classes are often presented with their attributes elided, to avoid clutter. The fully described class can usually be found in a related diagram. A class presented with an empty section at the bottom of the icon indicates that there are no attributes other than those that are visualized using associations.

1.2.3.1 Class Properties

Generally, a class property can be shown in a UML diagram as either an attribute or an association (i.e., the distinction between attributes and associations is somewhat subjective). In order to make the size of UML diagrams in the specifications manageable, we have chosen to capture most properties as attributes and to capture only higher level properties as associations, especially in the main top-level component diagrams. In particular, we will always capture properties of UML data types as attributes. For example, properties of a class that are identifiers, titles, and timestamps will be represented as attributes.

1.2.3.2 Diagram Icons and Arrow Types

Diagram icons are used in a UML diagram to indicate whether a shape is a class, enumeration or data type, and decorative icons are used to indicate whether an element is an attribute of a class or an enumeration literal. In addition, two different arrow styles indicate either a directed association relationship (regular arrowhead) or a generalization relationship (triangle-shaped arrowhead). The icons and arrow styles we use are shown and described in **Table 1-1**.

Table 1-1. UML diagram icons

Icon	Description
▤	This diagram icon indicates a class. If the name is in italics, it is an abstract class.
‹E›	This diagram icon indicates an enumeration.
‹D›	This diagram icon indicates a data type.
▤●	This decorator icon indicates an attribute of a class. The green circle means its visibility is public. If the circle is red or yellow, it means its visibility is private or protected.
⊟	This decorator icon indicates an enumeration literal.
⟶	This arrow type indicates a directed association relationship.
—▷	This arrow type indicates a generalization relationship.

1.2.3.3 Color Coding

The shapes of the UML diagrams are color coded to indicate the data model associated with a class. The colors used in the Core specification are illustrated via exemplars in **Figure 1-2**. The overarching Core and Common data models, use the same light blue color coding.

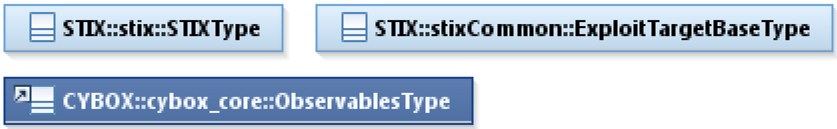

STIX::stix::STIXType
STIX::stixCommon::ExploitTargetBaseType
CYBOX::cybox_core::ObservablesType

Figure 1-2. Data model color coding

1.2.4 Property Table Notation

Throughout Section **3**, tables are used to describe the properties of each data model class. Each property table consists of a column of names to identify the property, a type column to reflect the datatype of the property, a multiplicity column to reflect the allowed number of occurrences of the property, and a description column that describes the property. Package prefixes are provided for classes outside of the Core data model (see Section **1.2.2**).

Note that if a class is a specialization of a superclass, only the properties that constitute the specialization are shown in the property table (i.e., properties of the superclass will not be shown). However, details of the superclass may be shown in the UML diagram.

1.2.5 Property and Class Descriptions

Each class and property defined in STIX is described using the format, “The X property verb Y.” For example, in the specification for the STIX Indicator, we write, “The id property specifies a globally unique identifier for the kill chain instance.” In fact, the verb “specifies” could have been replaced by any number of alternatives: “defines,” “describes,” “contains,” “references,” etc.

However, we thought that using a wide variety of verb phrases might confuse a reader of a specification document because the meaning of each verb could be interpreted slightly differently. On the other hand, we didn’t want to use a single, generic verb, such as “describes,” because although the different verb choices may or may not be meaningful from an implementation standpoint, a distinction could be useful to those interested in the modeling aspect of STIX.

Consequently, we have chosen to use the three verbs, defined as follows, in class and property descriptions:

Verb	STIX Definition
captures	Used to record and preserve information without implying anything about the structure of a class or property. Often used for properties that encompass general content. This is the least precise of the three verbs.
	Examples: The Source property characterizes the source of the sighting information. Examples of details captured include identitifying characteristics, time-related attributes, and a list of the tools used to collect the information. The Description property captures a textual description of the Indicator.
characterizes	Describes the distinctive nature or features of a class or property. Often used to describe classes and properties that themselves comprise one or more other properties.

	Examples: The `Confidence` property characterizes the level of confidence in the accuracy of the overall content captured in the Incident. The `ActivityType` class characterizes basic information about an activity a defender might use in response to a Campaign.
specifies	Used to clearly and precisely identify particular instances or values associated with a property. Often used for properties that are defined by a controlled vocabulary or enumeration; typically used for properties that take on only a single value.
	Example: The `version` property specifies the version identifier of the STIX Campaign data model used to capture the information associated with the Campaign.

1.3 Terminology

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 **[RFC2119]**.

1.4 Normative References

[RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, BCP 14, RFC 2119, March 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2119.txt.

1.5 Non-Normative References

[V111] DRAFT STIX specification documents for version 1.1.1. (n.d.). [Online]. Available: https://github.com/STIXProject/specifications/tree/master. Accessed Aug. 24, 2015.

2 Background Information

In this section, we provide high level information about the Core data model that is necessary to fully understand the specification details given in Section **3**.

As will be explicitly detailed in Section **3**, the STIX Core data model leverages all nine top-level component data models. **Figure 2-1** illustrates the concept of a STIX Package, which acts as an *envelope* for the other top-level constructs in a STIX document. As stated in Section **1.1**, each of these components is defined in a separate specification document.

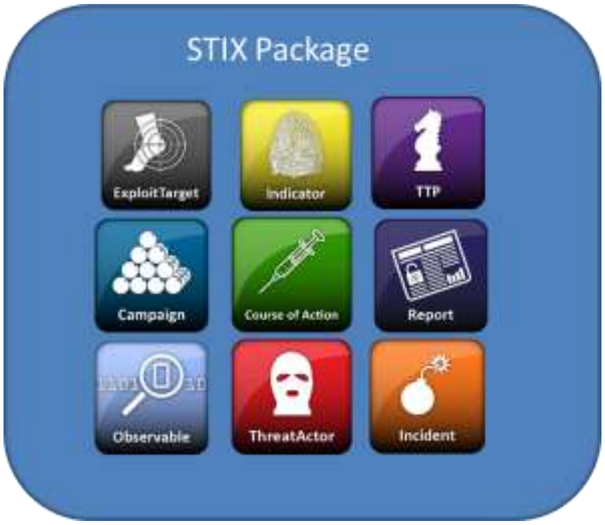


Figure 2-1. A STIX Package

Because a STIX Package is simply a container to carry content, the fact that construct instances appear in the same package does not mean that they are related in any way. As a deprecated capability, the STIX Package Header may characterize general information such as title, description, and package intent. If these deprecated fields are used, they give context to the collection of objects contained in the package as defined in the STIX 1.1.1 specification **[V111]**.

2.1 Component Data Models

Individual component data models define objects specific to each top-level STIX component construct: Observable; Indicator; Incident; Tactics, Techniques, and Procedures (TTPs); Exploit Target; Course of Action (COA); Campaign; Threat Actor, and Report. These data models each provide the capability to fully express information about their targeted conceptual area. In the STIX framework, they are all optional and may be used separately or in concert, as appropriate, using whichever components and architectural relationships that are relevant for a given use case.

In the subsections below, a brief description is given for each component data model as well as a reference to the data model's individual specification document.

2.1.1 Observable

A STIX Observable (as defined with the CybOX Language) represents stateful properties or measurable events pertinent to the operation of computers and networks. Implicit in this is a practical need for descriptive capability of two forms of observables: “observable instances” and “observable patterns.” Observable instances represent actual specific observations that took place in the cyber domain. The property details of this observation are specific and unambiguous. Observable patterns represent conditions for a potential observation that may occur in the future or may have already occurred and exists in a body of observable instances. These conditions may be anything from very specific concrete patterns that would match very specific observable instances to more abstract generalized patterns that have the potential to match against a broad range of potential observable instances.

2.1.2 Indicator

A STIX Indicator conveys specific Observable patterns combined with contextual information intended to represent artifacts and/or behaviors of interest within a cyber security context. Please see *STIX Version 1.2.1 Part 4: Indicator* for details.

2.1.3 Incident

A STIX Incident corresponds to sets of related security events affecting an organization, along with information discovered or decided during an incident response investigation. Please see *STIX Version 1.2.1 Part 6: Incident* for details.

2.1.4 Tactics, Techniques and Procedures (TTP)

A STIX Tactics, Techniques, and Procedures (TTP) is used to represent the behavior or modus operandi of cyber adversaries. Please see *STIX Version 1.2.1 Part 5: TTP* for details.

2.1.5 Campaign

A STIX Campaign represents a set of TTPs, Incidents, or Threat Actors that together express a common intent or desired effect. Please see *STIX Version 1.2.1 Part 8: Campaign* for details.

2.1.6 Threat Actor

A STIX Threat Actor is a characterization of a malicious actor (or adversary) representing a cyber attack threat including presumed intent and historically observed behavior. Please see *STIX Version 1.2.1 Part 7: Threat Actor* for details.

2.1.7 Exploit Target

A STIX Exploit Target conveys information about a vulnerability, weakness, or misconfiguration in software, systems, networks, or configurations that may be targeted for exploitation by an adversary. Please see *STIX Version 1.2.1 Part 10: Exploit Target* for details.

2.1.8 Course of Action (COA)

A STIX Course of Action (COA) is used to convey information about courses of action that may be taken either in response to an attack or as a preventative measure prior to an attack. Please see *STIX Version 1.2.1 Part 9: Course of Action* for details.

2.1.9 Report

A STIX Report construct defines a contextual wrapper for a grouping of STIX content, which could include content specified using any of the other nine top-level constructs, even including other related Reports. Please see *STIX Version 1.2.1 Part 11: Report* for details.

3 STIX[TM] Core Data Model

The primary class of the STIX Core package is the `STIXType` class, which defines a bundle of information characterized in the Structured Threat Information Expression (STIX) language. We refer to this bundle of information as a “STIX Package”[2].

The properties of the `STIXType` class, are illustrated in the UML diagram given in **Figure 3-1** on page 14.

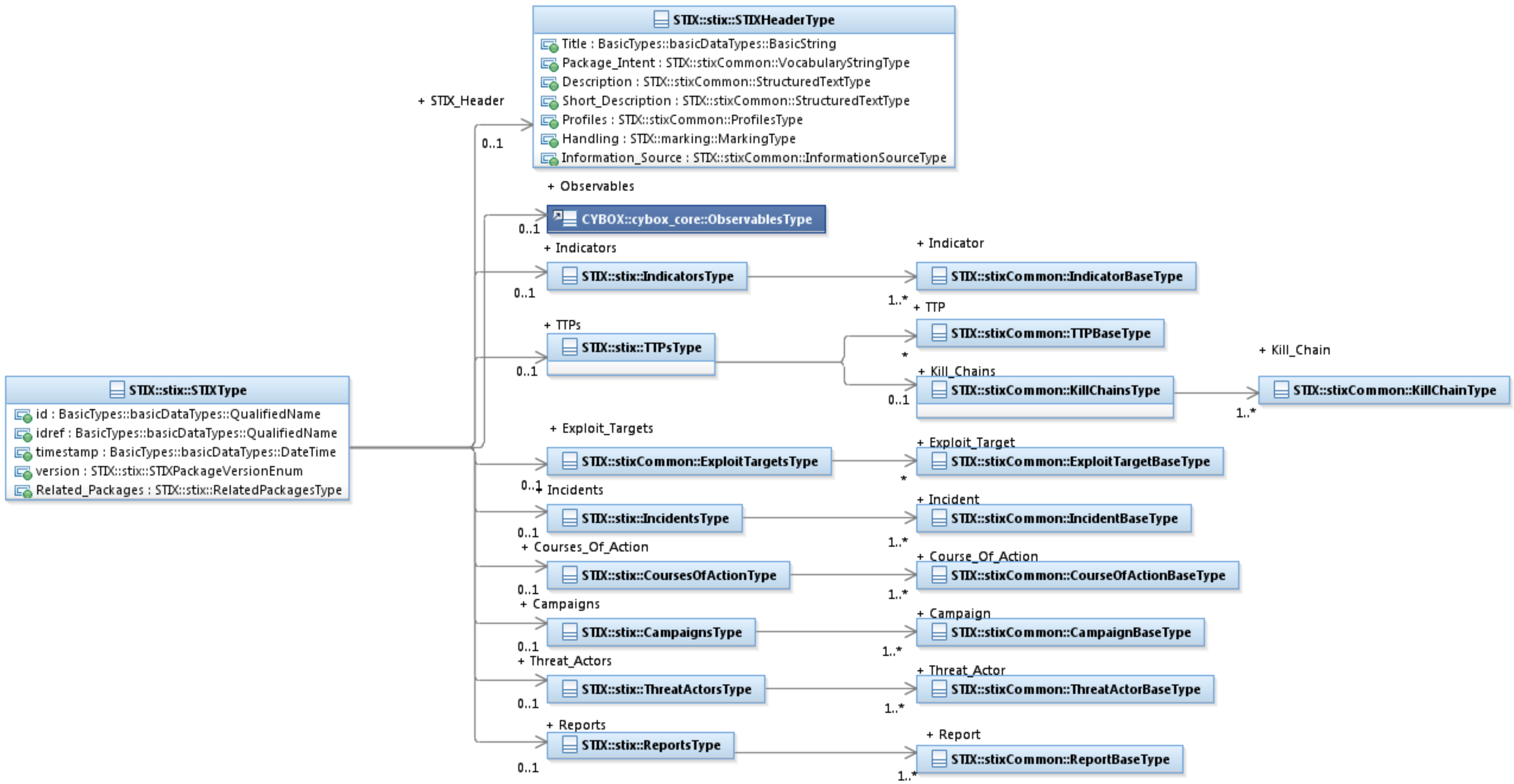

STIX::stix::STIXHeaderType
Title : BasicTypes::basicDataTypes::BasicString
Package_Intent : STIX::stixCommon::VocabularyStringType
Description : STIX::stixCommon::StructuredTextType
Short_Description : STIX::stixCommon::StructuredTextType
Profiles : STIX::stixCommon::ProfilesType
Handling : STIX::marking::MarkingType
Information_Source : STIX::stixCommon::InformationSourceType
+ STIX_Header
0..1
+ Observables
CYBOX::cybox_core::ObservablesType
0..1
+ Indicators
STIX::stix::IndicatorsType
0..1
+ Indicator
STIX::stixCommon::IndicatorBaseType
1..*
+ TTP
+ TTPs
STIX::stix::TTPsType
0..1
STIX::stixCommon::TTPBaseType
*
+ Kill_Chains
STIX::stixCommon::KillChainsType
0..1
+ Kill_Chain
STIX::stixCommon::KillChainType
1..*
STIX::stix::STIXType
id : BasicTypes::basicDataTypes::QualifiedName
idref : BasicTypes::basicDataTypes::QualifiedName
timestamp : BasicTypes::basicDataTypes::DateTime
version : STIX::stix::STIXPackageVersionEnum
Related_Packages : STIX::stix::RelatedPackagesType
+ Exploit_Targets
STIX::stixCommon::ExploitTargetsType
0..1
+ Exploit_Target
STIX::stixCommon::ExploitTargetBaseType
*
+ Incidents
STIX::stix::IncidentsType
0..1
+ Incident
STIX::stixCommon::IncidentBaseType
1..*
+ Courses_Of_Action
STIX::stix::CoursesOfActionType
0..1
+ Course_Of_Action
STIX::stixCommon::CourseOfActionBaseType
1..*
+ Campaigns
STIX::stix::CampaignsType
0..1
+ Campaign
STIX::stixCommon::CampaignBaseType
1..*
+ Threat_Actors
STIX::stix::ThreatActorsType
0..1
+ Threat_Actor
STIX::stixCommon::ThreatActorBaseType
1..*
+ Reports
STIX::stix::ReportsType
0..1
+ Report
STIX::stixCommon::ReportBaseType
1..*

Figure 3-1. UML diagram of the `STIXType` class

Table 3-1. Properties of The `STIXType` class

Name	Type	Multiplicity	Description

id	`BasicDataTypes:`
	`QualifiedName`
		0..1	The `id` property specifies a globally unique identifier for the STIX Package.

idref	`BasicDataTypes:`
	`QualifiedName`
		0..1	The `idref` property specifies an identifier reference to a STIX Package specified elsewhere. When the `idref` property is used, the id property MUST NOT also be specified and the other properties of the `STIXType` class SHOULD NOT hold any content.
			DEPRECATED: This property is deprecated and will be removed in the next major version of STIX. Its use is strongly discouraged except for legacy applications.

timestamp	`BasicDataTypes:`
	`DateTime`
		0..1	The `timestamp` property specifies a timestamp for the definition of a specific version of a STIX Package. When used in conjunction with the `id` property, this property specifies the definition time for the specific version of the STIX Package. When used in conjunction with the `idref` property, this property specifies a reference to a specific version of a STIX Package defined elsewhere. This property has no defined semantic meaning if used in the absence of either the `id` or `idref` properties.

version	`STIXPackageVersionEnum`	0..1	The `version` property specifies the version identifier of the STIX Core data model for STIX v1.2.1 used to capture the information associated with the STIX Package.
STIX_Header	`STIXHeaderType`	0..1	The `STIX_Header` property characterizes the metadata for this package of STIX content.
Observables	`cybox:ObservablesType`	0..1	The `Observables` property specifies a set of one or more cyber observables.
Indicators	`IndicatorsType`	0..1	The `Indicators` property specifies a set of one or more cyber threat Indicators.

TTPs	TTPsType	0..1	The TTPs property specifies a set of one or more cyber threat adversary Tactics, Techniques or Procedures (TTPs), or Kill Chains.
Exploit_Targets	stixCommon:ExploitTargetsType	0..1	The Exploit_Targets property specifies a set of zero or more potential targets for exploitation.
Incidents	IncidentsType	0..1	The Incidents property specifies a set of one or more cyber threat Incidents.
Courses_Of_Action	CoursesOfActionType	0..1	The CoursesOfActions property specifies a set of one or more Courses of Action that could be taken in regard to one of more cyber threats.
Campaigns	CampaignsType	0..1	The Campaigns property specifies a set of one or more Campaigns.
Threat_Actors	ThreatActorsType	0..1	The ThreatActors property specifies a set of one or more Threat Actors.
Reports	ReportsType	0..1	The Reports property specifies a set of one or more Reports.
Related_Packages	RelatedPackagesType	0..1	The Related_Packages property specifies a set of one or more Packages which may be relevant to this STIX Package.

DEPRECATION NOTICE: The use of the @idref attribute on any instance at the top level of the content aggregation classes is deprecated and will be removed in the next major version of STIX. Its use is strongly discouraged except for legacy applications. Instances in these content aggregation classes should only be embedded, not referenced.

3.1 STIXPackageVersionType Enumeration

The STIXPackageVersionType enumeration is an inventory of all versions of the STIX Core data model for STIX Version 1.2.1. The enumeration literals are given in **Table 3-2**.

Table 3-2. Literals of the STIXPackageVersionType enumeration

Enumeration Literal	Description
stix-1.2.1	STIX Core data model for STIX v1.2.1

3.2 STIXHeaderType Class

The `STIXHeaderType` class provides a structure for characterizing a package of STIX content.

The properties of the `STIXHeaderType` class are given in **Table 3-3**.

Table 3-3. Properties of the `STIXHeaderType` class

Name	Type	Multiplicity	Description
Title	`basicDataTypes:BasicString`	0..1	The `Title` property captures a title for the STIX Package and reflects what the content producer thinks the Package as a whole should be called. The Ti`t`le property is typically used by humans to reference a particular Package; however, it is not suggested for correlation. DEPRECATED: This property is deprecated and will be removed in the next major version of STIX. Its use is strongly discouraged except for legacy applications.

Package_Intent	`stixCommon:`
	`VocabularyStringType`
		0..*	The `Package_Intent` property specifies the intended purpose(s) or use(s) for The STIX Package. Examples of potential purposes are *phishing*, *exploit characterization* and *malware samples* (these specific values are only provided to help explain the property: they are neither recommended types nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary value or may constrain the set of possible values by referencing an externally-defined vocabulary or leveraging a formally defined vocabulary extending from the `stixCommon:ControlledVocabularyStringType` class. The STIX default vocabulary class for use in this property is *'PackageIntentVocab-1.0'*.
			DEPRECATED: This property is deprecated and will be removed in the next major version of STIX. Its use is strongly discouraged except for legacy applications.

Description	stixCommon:StructuredTextType	0..*	The Description property captures a textual description of the STIX Package. Any length is permitted. Optional formatting is supported via the structuring_format property of the StructuredTextType class. DEPRECATED: This property is deprecated and will be removed in the next major version of STIX. Its use is strongly discouraged except for legacy applications.
Short_Description	stixCommon:StructuredTextType	0..*	The Short_Description property captures a short textual description of the STIX Package. This property is secondary and should only be used if the Description property is already populated and another, shorter description is available. DEPRECATED: This property is deprecated and will be removed in the next major version of STIX. Its use is strongly discouraged except for legacy applications.
Profiles	stixCommon:ProfilesType	0..1	The Profiles property specifies a set of one or more profiles that the content of the STIX Package conforms to.
Handling	marking:MarkingType	0..1	The Handling property specifies the appropriate data handling markings for the properties of this STIX Package. The marking scope is limited to the STIX Package and the content it contains. Note that data handling markings can also be specified at a higher level.
Information_Source	stixCommon:InformationSourceTyp	0..1	The Information_Source property characterizes the source of the STIX Package and all of its contained information. Examples of details captured include identitifying characteristics, time-related attributes, and a list of the tools used to collect the information.

3.3 Content Aggregation Types

Each component type has an associated aggregation class that has one main property – a set of instances of that component type. The aggregation class for Observables, cybox_core:ObservablesType, is defined in *STIX Version 1.2.1 Part 3: Core*.

3.3.1 CampaignsType Class

The CampaignsType class specifies a set of one or more cyber threat Campaigns.

The properties of the CampaignsType class are given in **Table 3-4**.

Table 3-4. Properties of the `CampaignsType` class

Name	Type	Multiplicity	Description
Campaign	`stixCommon:CampaignBaseType`	1..*	The `Campaign` property characterizes a cyber threat Campaign. The `stixCommon:CampaignBaseType` class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement a Campaign is the `campaign:CampaignType` class defined in *STIX Version 1.2.1 Part 8: Campaign*. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation.

3.3.2 CoursesOfActionType Class

The `CoursesOfActionType` class specifies a set of one or more actions that could be taken in regard to cyber threats.

The properties of the `CoursesOfActionType` class are given in **Table 3-5**.

Table 3-5. Properties of the `CoursesOfActionType` class

Name	Type	Multiplicity	Description

Course_Of_Action	`stixCommon:`
	`CourseOfActionBaseType`
		1..*	The `Course_Of_Action` property characterizes a Course of Action that could be taken in regard to one of more cyber threats. The `stixCommon:CourseOfActionBaseType` class is a minimal base class that is intended to be extended. The default and strongly RECOMMENDED class to fully implement a Course of Action is the `coa:CourseOfActionType` class defined in *STIX Version 1.2.1 Part 9: Course of Action*. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation.

3.3.3 IncidentsType Class

The `IncidentsType` class specifies a set of one or more cyber threat Incidents.

The properties of the `IncidentsType` class are given in **Table 3-6**.

Table 3-6. Properties of the `IncidentsType` class

Name	Type	Multiplicity	Description
Incident	stixCommon: IncidentBaseType	1..*	The Incident property characterizes a cyber threat Incident. The stixCommon:IncidentBaseType class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement an Incident is the incident:IncidentType class defined in *STIX Version 1.2.1 Part 6: Incident*. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation.

3.3.4 IndicatorsType Class

The IndicatorsType class specifies a set of one or more cyber threat Indicators.

The properties of the IndicatorsType class are given in **Table 3-7**.

Table 3-7. Properties of the IndicatorsType class

Name	Type	Multiplicity	Description
Indicator	stixCommon: IndicatorBaseType	1..*	The Indicator property characterizes a cyber threat Indicator. The stixCommon:IndicatorBaseType class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement an Indicator is the indicator:IndicatorType class defined in *STIX Version 1.2.1 Part 4: Indicator*. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation.

3.3.5 ThreatActorsType Class

The ThreatActorsType class specifies a set of one or more cyber Threat Actors.

The properties of the ThreatActorsType class are given in **Table 3-8**.

Table 3-8. Properties of the ThreatActorsType class

Name	Type	Multiplicity	Description

Threat_Actor	`stixCommon:`
	`ThreatActorBaseType`
		1..*	The `ThreatActor` property characterizes a cyber Threat Actor. The `stixCommon:ThreatActorBaseType` class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement an ThreatActor is the `ta:ThreatActorType` class defined in *STIX Version 1.2.1 Part 7: Threat Actor*. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation.

3.3.6 TTPsType Class

The `TTPsType` class specifies a set of one or more cyber threat TTPs or Kill Chains.

The properties of the `TTPsType` class are given in **Table 3-9**.

Table 3-9. Properties of the `TTPsType` class

Name	Type	Multiplicity	Description
TTP	`stixCommon:TTPBaseType`	0..*	The `TTP` property characterizes a cyber threat adversary Tactic, Technique or Procedure (TTP). The `stixCommon:TTPBaseType` class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement a TTP is the `ttp:TTPType` class defined in *STIX Version 1.2.1 Part 5: TTP*. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation.
Kill_Chains	`stixCommon:KillChainsType`	0..1	A cyber kill chain is a phase-based model to describe the stages of an attack. The `Kill_Chains` property specifies a set of one or more specific kill chain definitions. The `kill_chain` property is further defined in the STIX Common specification document. Note that kill chains may also be defined using the `Kill_Chains` property of the TTP `TTPType` class, which is equivalent to this property. Suggested practice is to use the TTP `TTPType Kill_Chains` property (rather than this property) to define a kill chain.

3.3.7 ReportsType

The `ReportsType` class specifies a set of one or more cyber threat Reports.

The properties of the `ReportsType` class are given in **Table 3-10**.

Table 3-10. Properties of `ReportsType` class

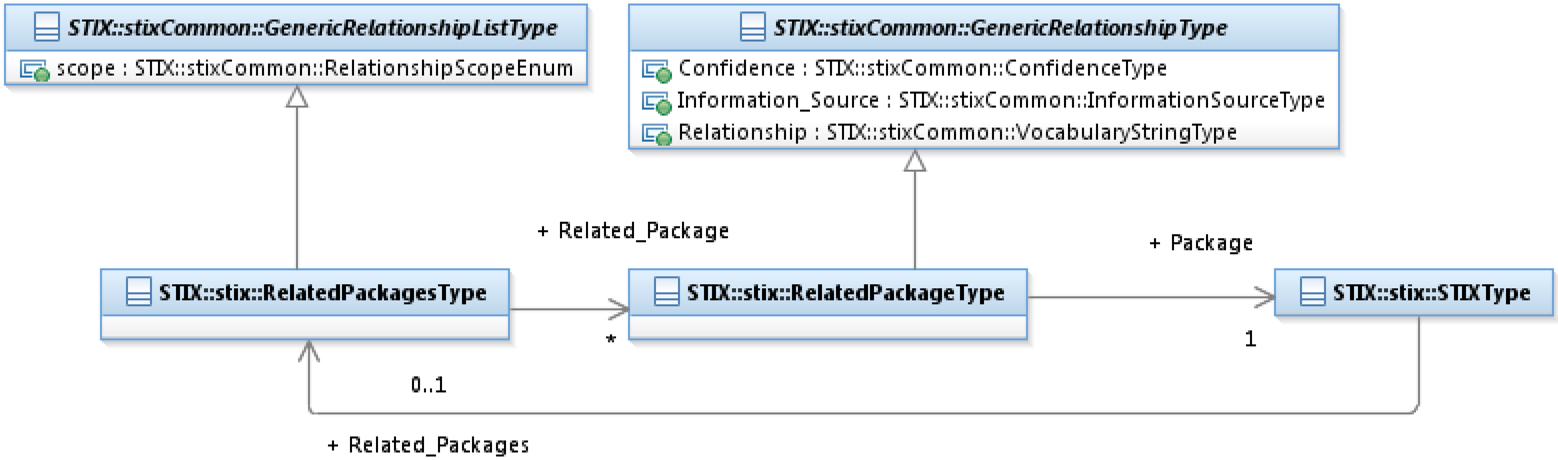
Name	Type	Multiplicity	Description

Report	`stixCommon:`
	`ReportBaseType`
		1..*	The `Report` property characterizes a cyber threat Report. The `stixCommon:Report` BaseType class is a minimal base class that is intended to be extended. The default and strongly recommended class to fully implement a Report is the `report:ReportType` class defined in *STIX Version 1.2.1 Part 11: Report*. Base classes are used to minimize interdependence between STIX components, not to enable or encourage conflicting syntactic variation.

3.4 RelatedPackagesType Class

The `RelatedPackagesType` class specifies a set of one or more STIX Package related to this STIX Package. It extends the `GenericRelationShipListType` superclass defined in the STIX Common data model, which specifies the scope (whether the elements of the set are related individually or as a group).

The UML diagram corresponding to the `RelatedPackagesType` class is shown in **Figure 3-2**, and the specialized properties are shown in **Table 3-11**.



STIX::stixCommon::GenericRelationshipListType
scope : STIX::stixCommon::RelationshipScopeEnum
STIX::stixCommon::GenericRelationshipType
Confidence : STIX::stixCommon::ConfidenceType
Information_Source : STIX::stixCommon::InformationSourceType
Relationship : STIX::stixCommon::VocabularyStringType
+ Related_Package
+ Package
STIX::stix::RelatedPackagesType
STIX::stix::RelatedPackageType
STIX::stix::STIXType
*
1
0..1
+ Related_Packages

Figure 3-2. UML diagram for `RelatedPackagesType` class

In addition to being a property of the `STIXType` class, `Related_Packages` is a property of all of the top-level component types.

*The property table given in **Table 3-11** corresponds to the UML diagram shown in **Figure 3-2**.*

Table 3-11. Properties of `RelatedPackagesType` class

Name	**Type**	**Multiplicity**	**Description**
Related_Package	`RelatedPackageType`	0..*	The `Related_Package` property characterizes a relationship to one or more other STIX Packages.

3.4.1 RelatedPackageType Class

The `RelatedPackageType` class identifies or characterizes the relationship of STIX Package to another.

The properties of the `RelatedPackageType` class are given in **Table 3-12**.

Table 3-12. Properties of `RelatedPackageType` class

Name	**Type**	**Multiplicity**	**Description**
Package	`STIXType`	1	The `Package` property captures or references a STIX Package related to this STIX Package.

4 Conformance

Implementations have discretion over which parts (components, properties, extensions, controlled vocabularies, etc.) of STIX they implement (e.g., Indicator/Suggested_COAs).

[1] Conformant implementations must conform to all normative structural specifications of the UML model or additional normative statements within this document that apply to the portions of STIX they implement (e.g., Implementers of the entire TTP component must conform to all normative structural specifications of the UML model or additional normative statements within this document regarding the TTP component).

[2] Conformant implementations are free to ignore normative structural specifications of the UML model or additional normative statements within this document that do not apply to the portions of STIX they implement (e.g., Non-implementers of any particular properties of the TTP component are free to ignore all normative structural specifications of the UML model or additional normative statements within this document regarding those properties of the TTP component).

The conformance section of this document is intentionally broad and attempts to reiterate what already exists in this document. The STIX 1.2 Specifications, which this specification is based on, did not have a conformance section. Instead, the STIX 1.2 Specifications relied on normative statements and the non-mandatory implementation of STIX profiles. STIX 1.2.1 represents a minimal change from STIX 1.2, and in that spirit no requirements have been added, modified, or removed by this section.

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Dean Thompson, Australia and New Zealand Banking Group (ANZ Bank)
Bret Jordan, Blue Coat Systems, Inc.
Adnan Baykal, Center for Internet Security (CIS)
Jyoti Verma, Cisco Systems
Liron Schiff, Comilion (mobile) Ltd.
Jane Ginn, Cyber Threat Intelligence Network, Inc. (CTIN)
Richard Struse, DHS Office of Cybersecurity and Communications (CS&C)
Marlon Taylor, DHS Office of Cybersecurity and Communications (CS&C)
David Eilken, Financial Services Information Sharing and Analysis Center (FS-ISAC)
Sarah Brown, Fox-IT
Ryusuke Masuoka, Fujitsu Limited
Eric Burger, Georgetown University
Jason Keirstead, IBM
Paul Martini, iboss, Inc.
Jerome Athias, Individual
Terry MacDonald, Individual
Alex Pinto, Individual
Patrick Maroney, Integrated Networking Technologies, Inc.
Wouter Bolsterlee, Intelworks BV
Joep Gommers, Intelworks BV
Sergey Polzunov, Intelworks BV
Rutger Prins, Intelworks BV
Andrei Sîrghi, Intelworks BV
Raymon van der Velde, Intelworks BV
Jonathan Baker, MITRE Corporation
Sean Barnum, MITRE Corporation
Desiree Beck, MITRE Corporation
Mark Davidson, MITRE Corporation
Ivan Kirillov, MITRE Corporation
Jon Salwen, MITRE Corporation
John Wunder, MITRE Corporation
Mike Boyle, National Security Agency
Jessica Fitzgerald-McKay, National Security Agency
Takahiro Kakumaru, NEC Corporation
John-Mark Gurney, New Context Services, Inc.
Christian Hunt, New Context Services, Inc.
Daniel Riedel, New Context Services, Inc.
Andrew Storms, New Context Services, Inc.
John Tolbert, Queralt, Inc.
Igor Baikalov, Securonix
Bernd Grobauer, Siemens AG
Jonathan Bush, Soltra
Aharon Chernin, Soltra
Trey Darley, Soltra
Paul Dion, Soltra
Ali Khan, Soltra
Natalie Suarez, Soltra
Cedric LeRoux, Splunk Inc.
Brian Luger, Splunk Inc.

Crystal Hayes, The Boeing Company
Brad Butts, U.S. Bank
Mona Magathan, U.S. Bank
Adam Cooper, United Kingdom Cabinet Office
Mike McLellan, United Kingdom Cabinet Office
Chris O'Brien, United Kingdom Cabinet Office
Julian White, United Kingdom Cabinet Office
Anthony Rutkowski, Yaana Technologies, LLC

The authors would also like to thank the larger STIX Community for its input and help in reviewing this document.

Appendix B. Revision History

Revision	Date	Editor	Changes Made
wd01	21 August 2015	Sean Barnum Desiree Beck Aharon Chernin Rich Piazza	Initial transfer to OASIS template

Notes

[1] The CybOX^{TM} Observable data model is actually defined in the CybOX Language, not in STIX.

[2] Throughout this section, a “STIX Package” denotes an object of type `STIXType` class.