STIX[TM] Version 1.2.1. Part 10: Exploit Target

Committee Specification Draft 01 / Public Review Draft 01

06 November 2015

Specification URIs

This version:

http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part10-exploit-target/stix-v1.2.1-csprd01-part10-exploit-target.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part10-exploit-target/stix-v1.2.1-csprd01-part10-exploit-target.html
http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part10-exploit-target/stix-v1.2.1-csprd01-part10-exploit-target.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part10-exploit-target.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part10-exploit-target.html
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part10-exploit-target.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/csprd01/part1-overview/stix-v1.2.1-csprd01-part1-overview.html
- *STIX Version 1.2.1. Part 2: Common*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part2-common/stix-v1.2.1-csprd01-part2-common.html
- *STIX Version 1.2.1. Part 3: Core*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part3-core/stix-v1.2.1-csprd01-part3-core.html
- *STIX Version 1.2.1. Part 4: Indicator*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part4-indicator/stix-v1.2.1-csprd01-part4-indicator.html
- *STIX Version 1.2.1. Part 5: TTP*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part5-ttp/stix-v1.2.1-csprd01-part5-ttp.html
- *STIX Version 1.2.1. Part 6: Incident*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part6-incident/stix-v1.2.1-csprd01-part6-incident.html

- *STIX Version 1.2.1. Part 7: Threat Actor*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part7-threat-actor/stix-v1.2.1-csprd01-part7-threat-actor.html
- *STIX Version 1.2.1. Part 8: Campaign*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part8-campaign/stix-v1.2.1-csprd01-part8-campaign.html
- *STIX Version 1.2.1. Part 9: Course of Action*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part9-coa/stix-v1.2.1-csprd01-part9-coa.html
- *STIX Version 1.2.1. Part 10: Exploit Target* (this document). http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part10-exploit-target/stix-v1.2.1-csprd01-part10-exploit-target.html
- *STIX Version 1.2.1. Part 11: Report*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part11-report/stix-v1.2.1-csprd01-part11-report.html
- *STIX Version 1.2.1. Part 12: Default Extensions*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part12-extensions/stix-v1.2.1-csprd01-part12-extensions.html
- *STIX Version 1.2.1. Part 13: Data Marking*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part13-data-marking/stix-v1.2.1-csprd01-part13-data-marking.html
- *STIX Version 1.2.1. Part 14: Vocabularies*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part14-vocabularies/stix-v1.2.1-csprd01-part14-vocabularies.html
- *STIX Version 1.2.1. Part 15: UML Model*. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part15-uml-model/stix-v1.2.1-csprd01-part15-uml-model.html
- UML Model Serialization: http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/uml-model/

Related work:

This specification replaces or supersedes:

- *STIX™ 1.2 Exploit Target Specification (v1.2)* https://github.com/STIXProject/specifications/blob/version1.2/documents/pdf%20versions/STIX_ExploitTarget_Draft.pdf

This specification is related to:

- *CybOX™ Version 2.1.1.* Work in progress. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=cti-cybox
- *CybOX™ 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 Exploit Target construct, which conveys a vulnerability or weakness in software, systems, networks or configurations that is targeted for exploitation by the TTP of a Threat Actor.

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-Exploit-Target]

STIX[TM] Version 1.2.1. Part 10: Exploit Target. Edited by Sean Barnum, Desiree Beck, Aharon Chernin, and Rich Piazza. 06 November 2015. OASIS Committee Specification Draft 01 / Public Review Draft 01. http://docs.oasis-open.org/cti/stix/v1.2.1/csprd01/part10-exploit-target/stix-v1.2.1-csprd01-part10-exploit-target.html. Latest version: http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part10-exploit-target.html.

Notices

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

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. This document serves as the specification for the STIX Exploit Target data model.

As defined within the STIX language, an Exploit Target is a vulnerability or weakness in software, systems, networks or configurations that is targeted for exploitation by the TTP of a Threat Actor. Recognizing a lack of current standardized approaches for generalized characterizations, STIX leverages community knowledge and best practices to define a new Exploit Target structure for representing exploit target information. Portions of the Exploit Target structure use existing standardized approaches to characterize vulnerabilities, weaknesses, and configurations.

More explicitly, the identifier constructs from the Common Vulnerabilities and Exposures (CVE®) and the Open Source Vulnerability Database (OSVDB) are used to identify publicly disclosed vulnerabilities. The Common Vulnerability Reporting Framework (CVRF) format is used to capture a detailed, structured characterization of vulnerabilities not identified in CVE or OSVDB (this allows for the characterization of zero-day vulnerabilities). The identifier construct from the Common Weakness Enumeration (CWE™) is used to identify weaknesses, and the identifier construct from the Common Configuration Enumeration (CCE™) is used to identify configuration issues.

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 Section **1.4**. 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 Exploit Target data model specification details in Section **1** and conformance information in Section **1**.

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 top-level component 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 Exploit Target 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*.

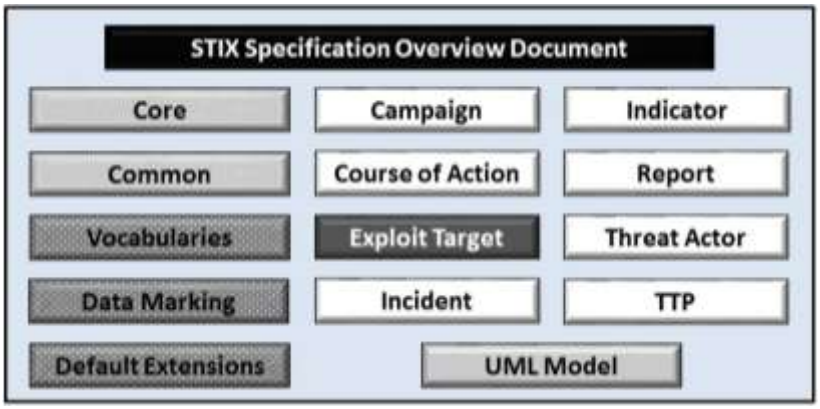


Figure 1-1. STIX[TM] Language v1.2.1 specification documents

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.) where the packages together compose the full STIX UML model. 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 Exploit Target 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 Indicator 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.

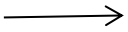
1.2.3.1 Class Properties

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.
	This diagram icon indicates an enumeration.
	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 Indicator specification are illustrated in **Figure 1-2**.

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 Exploit Target 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.

In addition, properties that are part of a "choice" relationship (e.g., Prop1 OR Prop2 is used but not both) will be denoted by a unique letter subscript (e.g., API_Call$_A$, Code$_B$) and single logic expression in the Multiplicity column. For example, if there is a choice of property `API_Call`$_A$ and `Code`$_B$, the expression "A(1)|B(0..1)" will indicate that the `API_Call` property can be chosen with multiplicity 1 or the `Code` property can be chosen with multiplicity 0 or 1.

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. http://www.ietf.org/rfc/rfc2119.txt.

2 Background Information

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

2.1 Exploit Target-Related Component Data Models

As will be explicitly detailed in Section **3**, a STIX Exploit Target leverages the Course of Action data model (as indicated by the outward-oriented arrow). **Figure 2-1** illustrates the relationship between the Exploit Target and the other core constructs. As stated in Section **1.1**, each of these components is defined in a separate specification document.

Figure 2-1. High level view of the Exploit Target data model

In this section, we give a high level summary of the relationship between the Exploit Target data model and the Course of Action to which an Exploit Target may refer. We also make note of the fact that the Exploit Target data model can be self-referential. Other relationships are defined in the specification of the component from which they originate.

- **Course of Action**

  A STIX Course of Action (COA) component 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. A Course of Action component captures a variety of information such as the Course of Action's objective, likely impact, efficacy, and cost. Please see *STIX Version 1.2.1 Part 9: Course of Action* for details.

  The Exploit Target data model references the Course of Action data model as a means to identify Courses of Actions that may be relevant in the mitigation of the Exploit Target.

- **Exploit Target**

  The Exploit Target data model is self-referential, enabling one Exploit Target to reference other Exploit Targets that are asserted to be related. Self-referential relationships between Exploit Targets may indicate general associativity or can be used to indicate relationships between different versions of the same Exploit Target.

3 STIX[TM] Exploit Target Data Model

The primary class of the STIX Exploit Target package is the `ExploitTargetType` class, which characterizes potential targets for exploitation by capturing characteristics of targeted victims that may make them vulnerable to attack. Similar to the primary classes of all the component data models in STIX, the `ExploitTargetType` class extends a base class defined in the STIX Common data model; more specifically, it specializes the `ExploitTargetBaseType` base class, which provides the essential identifier (`id`) and identifier reference (`idref`) properties.

The relationship between the `ExploitTargetType` class and the `ExploitTargetBaseType` base class, as well as the properties of the `ExploitTargetType` class, are illustrated in the UML diagram given in **Figure 3-1**.

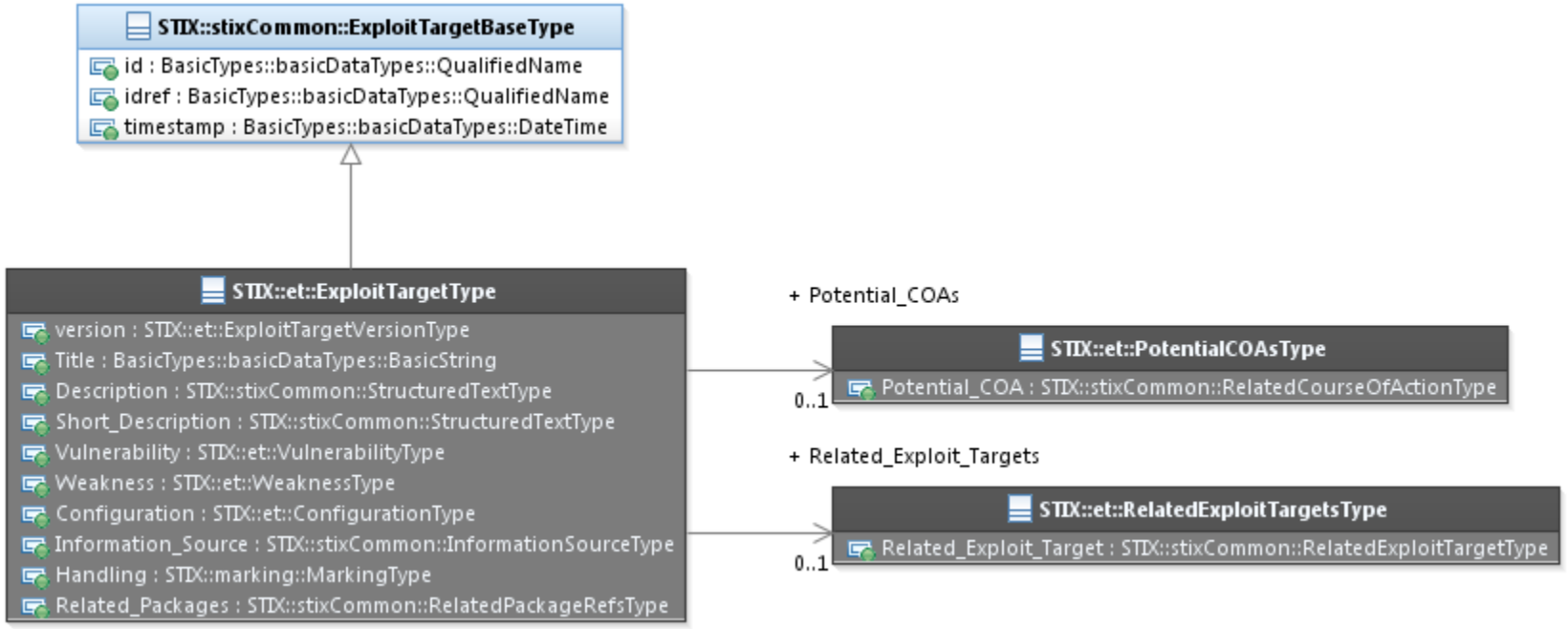


Figure 3-1. UML diagram of the `ExploitTargetType` class

The property table, which includes property descriptions and corresponds to the UML diagram given in **Figure 3-1**, is provided in **Table 3-1**.

Table 3-1. Properties of the `ExploitTargetType` class

Name	Type	Multiplicity	Description
version	`ExploitTargetVersionType`	0..1	The `version` property specifies the version identifier of the STIX Exploit Target data model for STIX v1.2.1 used to capture the information associated with the Exploit Target.
Title	`basicDataTypes:BasicString`	0..1	The `Title` property captures a title for the Exploit Target and reflects what the content producer thinks the Exploit Target as a whole should be called. The `Title` property is typically used by humans to reference a particular Exploit Target; however, it is not suggested for correlation.
Description	`stixCommon:StructuredTextType`	0..*	The `Description` property captures a textual description of the Exploit Target. Any length is permitted. Optional formatting is supported via the `structuring_format` property of the `StructuredTextType` class.
Short_Description	`stixCommon:StructuredTextType`	0..*	The `Short_Description` property captures a short textual description of the objective of this CourseOfAction. This property is secondary and should only be used if the `Description` property is already populated and another, shorter description is available.
Vulnerability	`VulnerabilityType`	0..*	The `Vulnerability` property characterizes a vulnerability that is a potential target for exploitation. Examples of information captured include a description of the vulnerability (in a structured or unstructured fashion), a CVE identifier, an OSVDB identifier, and CVSS information.
Weakness	`WeaknessType`	0..*	The `Weakness` property characterizes a weakness that is a potential target for exploitation. Examples of information captured include a description of the weakness and a CWE identifier.
Configuration	`ConfigurationType`	0..*	The `Configuration` property characterizes a configuration that is a potential target for exploitation. Examples of information captured include a description of

			the configuration issue and a CCE identifier.
Potential_COAs	`PotentialCOAsType`	0..1	The `Potential_COAs` property specifies a set of one or more Course of Actions that may be relevant for the remediation or mitigation of this Exploit Target.
Information_Source	`stixCommon:InformationSourceType`	0..1	The `Information_Source` property characterizes the source of the Exploit Target information. Examples of details captured include identitifying characteristics, time-related attributes, and a list of tools used to collect the information.
Handling	`marking:MarkingType`	0..1	The `Handling` property specifies the appropriate data handling markings for the properties of this Exploit Target. The marking scope is limited to the Exploit Target and the content it contains. Note that data handling markings can also be specified at a higher level.
Related_Exploit_Targets	`RelatedExploitTargetsType`	0..1	The `Related_Exploit_Targets` property specifies a set of one or more other Exploit Targets related to this Exploit Target.
Related_Packages	`stixCommon:RelatedPackagesRefsType`	0..1	The `Related_Packages` property specifies a set of one or more STIX Packages that are related to the Exploit Target.

3.1 ExploitTargetVersionType Enumeration

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

Table 3-2. Literals of the `ExploitTargetVersionType` enumeration

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

3.2 VulnerabilityType Class

The `VulnerabilityType` class characterizes an individual vulnerability. In addition to capturing basic information and references to vulnerability registries, this class is extensible to enable the structured description of a vulnerability. STIX v1.2.1 defines a default subclass to leverage the Common Vulnerability Reporting Format (CVRF) schema[2]: the `CVRF1.1InstanceType` class (see *STIX Version 1.2.1 Part 12: Default Extensions*).

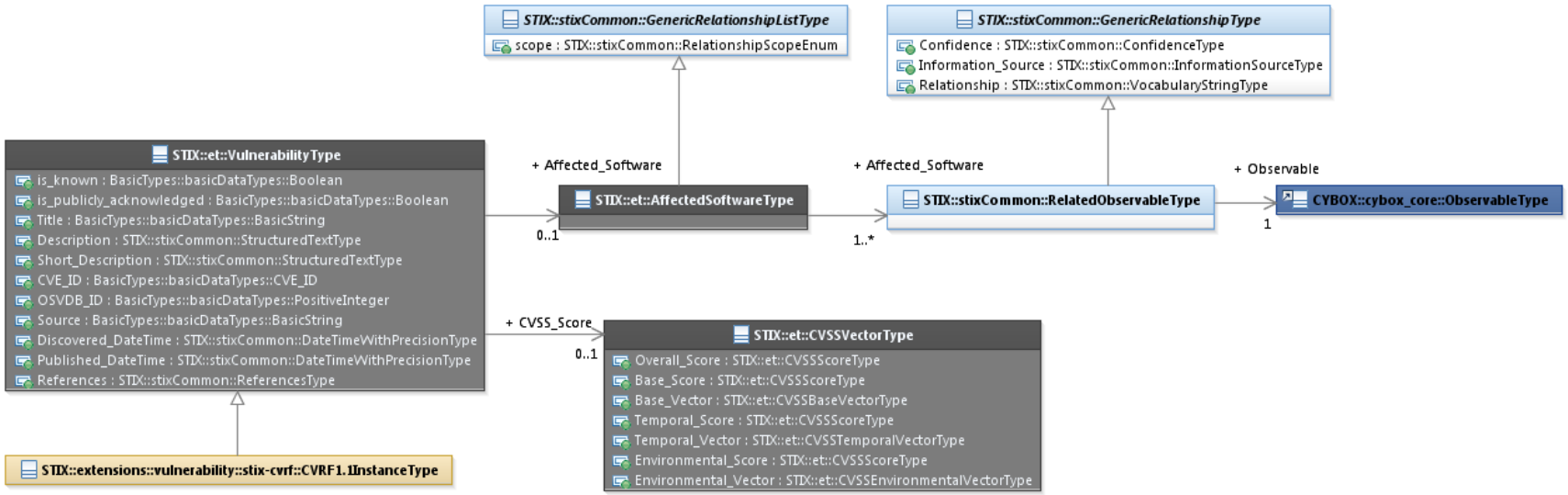


Figure 3-2. UML diagram of the `VulnerabilityType` class

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

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

Name	Type	Multiplicity	Description
is_known	`basicDataTypes:Boolean`	0..1	The `is_known` property specifies whether or not the vulnerability is known (i.e., not a 0-day) or unknown (i.e., a 0-day) at the time it is characterized.
is_publicly_acknowledged	`basicDataTypes:Boolean`	0..1	The `is_publicly_acknowledged` property specifies whether or not the vulnerability is publicly acknowledged by the vendor at the time it is characterized.

Title	basicDataTypes:String	0..1	The Title property captures a title for the vulnerability and reflects what the content producer thinks the vulnerability as a whole should be called. The Title property is typically used by humans to reference a particular vulnerability; however, it is not suggested for correlation.
Description	stixCommon: StructuredTextType	0..*	The Description property captures a textual description of the vulnerability. Any length is permitted. Optional formatting is supported via the structuring_format property of the StructuredTextType class.
Short_Description	stixCommon: StructuredTextType	0..*	The Short_Description property captures a short textual description of the vulnerability. This property is secondary and should only be used if the Description property is already populated and another, shorter description is available.
CVE_ID	basicDataTypes:CVE_ID	0..1	The CVE_ID property specifies a Common Vulnerability and Exposures (CVE) identifier for the vulnerability.
OSVDB_ID	basicDataTypes: PositiveInteger	0..1	The OSVDB_ID property specifies an Open Source Vulnerability Database (OSVDB) identifier for the vulnerability.
Source	basicDataTypes:String	0..1	The Source property captures a textual description or a URL of the original source of the vulnerability information.
CVSS_Score	CVSSVectorType	0..1	The CVSS_Score property captures the full Common Vulnerability Scoring System (CVSS) v2.0 base, temporal, and environmental vectors.
Discovered_DateTime	stixCommon: DateTimeWithPrecisionType	0..1	The Discovered_DateTime property specifies the date and time at which the vulnerability was discovered. To avoid ambiguity, all timestamps SHOULD include a specification of the time zone. In addition to specifying a date and time, the Date_Time property may also capture a precision property to specify the granularity with which the time should be considered, as specified by the DateTimePrecisionEnum enumeration (e.g., '*hour*,' '*minute*'). If omitted, the default precision is '*second*.' Digits in a timestamp that are beyond the specified precision SHOULD be zeroed out.

Published_DateTime	stixCommon: DateTimeWithPrecisionType	0..1	The Published_DateTime property specifies the date and time at which information about the vulnerability was published. To avoid ambiguity, all timestamps SHOULD include a specification of the time zone. In addition to specifying a date and time, the Date_Time property may also capture a precision property to specify the granularity with which the time should be considered, as specified by the DateTimePrecisionEnum enumeration (e.g., '*hour*,' '*minute*'). If omitted, the default precision is '*second*.' Digits in a timestamp that are beyond the specified precision SHOULD be zeroed out.
Affected_Software	AffectedSoftwareType	0..1	The Affected_Software property specifies a set of one or more software products that is affected by this vulnerability. It leverages the CybOX ObservableType class.
References	stixCommon:ReferencesType	0..1	The References property specifies a set of one or more related references associated with the vulnerability.

3.2.1 CVSSVectorType Class

The CVSSVectorType class characterizes Common Vulnerability Scoring System (CVSS) data associated with the vulnerability.

The property table for the CVSSVectorType class is given in **Table 3-4**.

Table 3-4. Properties of the CVSSVectorType class

Name	Type	Multiplicity	Description
Overall_Score	CVSSScoreType	0..1	The Overall_Score property specifies the CVSS 2.0 overall score. Note that this is not the same as the unadjusted CVSS base score, which should be specified in the Base_Score property.
Base_Score	CVSSScoreType	0..1	The Base_Score property specifies the unadjusted CVSS 2.0 base score.
Base_Vector	CVSSBaseVectorType	0..1	The Base_Vector property specifies the CVSS 2.0 base

			vector.
Temporal_Score	`CVSSScoreType`	0..1	The `Temporal_Score` property specifies the CVSS 2.0 temporal score.
Temporal_Vector	`CVSSTemporalVectorType`	0..1	The `Temporal_Vector` property specifies the CVSS 2.0 temporal vector.
Environmental_Score	`CVSSScoreType`	0..1	The `Environmental_Score` property specifies the CVSS 2.0 environmental score.
Environmental_Vector	`CVSSEnvironmentalVectorType`	0..1	The `Environmental_Vector` property specifies the CVSS 2.0 environmental vector.

3.2.1.1 CVSSScoreType Data Type

The `CVSSScoreType` data type specializes the `basicDataTypes:BasicString` data type by restricting it to the pattern: ((10)|[0-9])\.[0-9]

3.2.1.2 CVSSBaseVectorType Data Type

The `CVSSBaseVectorType` data type specializes the `basicDataTypes:BasicString` data type by restricting it to the pattern: AV:[LAN]/AC:[HML]/Au:[MSN]/C:[NPC]/I:[NPC]/A:[NPC]

3.2.1.3 CVSSTemporalVectorType Data Type

The `CVSSTemporalVectorType` data type specializes the `basicDataTypes:BasicString` data type by restricting it to the pattern: E:([UFH]|(POC)|(ND))/RL:([WU]|(OF)|(TF)|(ND))/RC:([C]|(UC)|(UR)|(ND))

3.2.1.4 CVSSEnvironmentalVectorType Data Type

The `CVSSEnvironmentalVectorType` data specializes the `basicDataTypes:BasicString` data type by restricting it to the pattern: CDP:([NLH]|(LM)|(MH)|(ND))/TD:([NLMH]|(ND))/CR:([LMH]|(ND))/IR:([LMH]|(ND))/AR:([LMH]|(ND))

3.2.2 AffectedSoftwareType Class

The `AffectedSoftwareType` class specifies a set of platforms and software that are affected by a vulnerability. 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 property table for the AffectedSoftwareType class is given in **Table 3-5**.

Table 3-5. Properties of the AffectedSoftwareType class

Name	Type	Multiplicity	Description
Affected_Software	stixCommon:RelatedObservableType	1..*	The Affected_Software property characterizes a single software product or platform affected by this vulnerability.

3.3 WeaknessType Class

The WeaknessType class characterizes a weakness as a potential Exploit Target.

The property table for the WeaknessType class is given in **Table 3-6**.

Table 3-6. Properties of the WeaknessType class

Name	Type	Multiplicity	Description
Description	stixCommon:StructuredTextType	0..*	The Description property captures a textual description of the weakness. Any length is permitted. Optional formatting is supported via the structuring_format property of the StructuredTextType class.
CWE_ID	basicDataTypes:CWE_ID	0..1	The CWE_ID property specifies a Common Weakness Enumeration (CWE) identifier for a particular weakness.

3.4 ConfigurationType Class

The ConfigurationType class characterizes a software or hardware configuration as a potential Exploit Target.

The property table for the WeaknessType class is given in **Table 3-7**.

Table 3-7. Properties of the ConfigurationType class

Name	Type	Multiplicity	Description

Description	stixCommon:StructuredTextType	0..*	The Description property captures a textual description of the configuration. Any length is permitted. Optional formatting is supported via the structuring_format property of the StructuredTextType class.
Short_Description	stixCommon:StructuredTextType	0..*	The Short_Description property captures a short textual description of the configuration. This property is secondary and should only be used if the Description property is already populated and another, shorter description is available.
CCE_ID	basicDataTypes:CCE_ID	0..1	The CCE_ID property specifies a Common Configuration Enumeration (CCE) identifier for a particular configuration item.

3.5 PotentialCOAsType Class

The PotentialCOAsType class specifies a set of one or more potential Course of Actions (COAs) for the Exploit Target. 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 PotentialCOAsType class is shown in **Figure 3-3**.

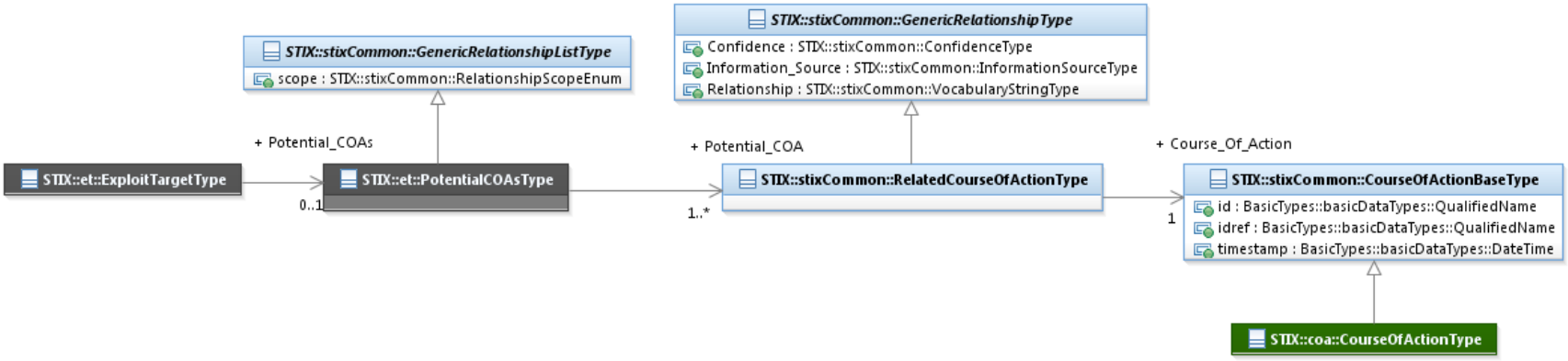


Figure 3-3. UML diagram of the PotentialCOAsType class

*The property table given in **Table 3-8** corresponds to the UML diagram shown in **Figure 3-3**.*

Table 3-8. Properties of the `PotentialCOAsType` class

Name	Type	Multiplicity	Description
Potential_COA	`stixCommon:RelatedCourseOfActionType`	1..*	The `Potential_COA` property specifies a Course of Action potentially relevant for the remediation or mitigation of this Exploit Target and characterizes this relevance relationship by capturing information such as the level of confidence that the Course of Action and the Exploit Target are related, the source of the relationship information, and the type of relationship.

3.6 RelatedExploitTargetsType Class

The `RelatedExploitTargetsType` class specifies a set of one or more other Exploit Targets asserted as related to this Exploit Target and therefore is a self-referential relationship. 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 `RelatedExploitTargetType` class is shown in **Figure 3-4**.

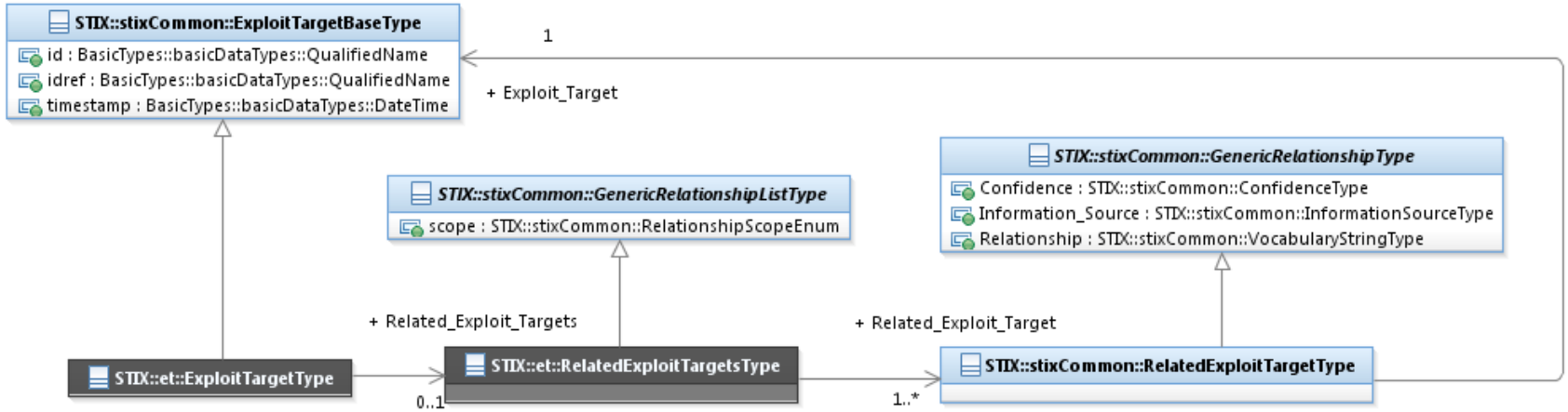


Figure 3-4. UML diagram of the `RelatedExploitTargetsType` class

The property table given in **Table 3-9** corresponds to the UML diagram shown in **Figure 3-4**.

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

Name	Type	Multiplicity	Description
Related_Exploit_Target	`stixCommon:RelatedExploitTargetType`	1..*	The `Related_Exploit_Target` property specifies another Exploit Target associated with this Exploit Target and characterizes the relationship between the Exploit Targets by capturing information such as the level of confidence that the Exploit Targets are related, the source of the relationship information, and type of the relationship. A relationship between Exploit Targets may represent assertions of general associativity or different versions of the same Exploit Target.

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 Observable data model is actually defined in the CybOX Language, not in STIX.

[2] There is no UML model defined for the CVRF; it is outside the scope of the STIX 1.2.1 specification.