STIX[TM] Version 1.2.1. Part 7: Threat Actor

Committee Specification Draft 01

06 November 2015

Specification URIs

This version:

http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part7-threat-actor/stix-v1.2.1-csd01-part7-threat-actor.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part7-threat-actor/stix-v1.2.1-csd01-part7-threat-actor.html
http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part7-threat-actor/stix-v1.2.1-csd01-part7-threat-actor.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part7-threat-actor.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part7-threat-actor.html
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part7-threat-actor.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*. 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* (this document). 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™ 1.2 Threat Actor Specification (v1.2).* https://github.com/STIXProject/specifications/blob/version1.2/documents/pdf%20versions/STIX_ThreatActor_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 Threat Actor construct, which captures characterizations of malicious actors (or adversaries) representing a cyber attack threat including presumed intent and historically observed behavior.

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-Threat-actor]

STIX[TM] Version 1.2.1. Part 7: Threat Actor. 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/part7-threat-actor/stix-v1.2.1-csd01-part7-threat-actor.html. Latest version: http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part7-threat-actor.html.

Notices

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 Threat Actor data model.

As defined within the STIX language, a Threat Actor construct captures characterizations of malicious actors (or adversaries) combined with contextual information, including presumed intent and historically observed behavior. In a structured sense, Threat Actors consist of a characterization of identity, suspected motivation, suspected intended effect, historically observed TTPs used, together with historical Campaigns and other Threat Actors believed associated with the Threat Actor.

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 Threat Actor 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 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 Threat Actor 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*.

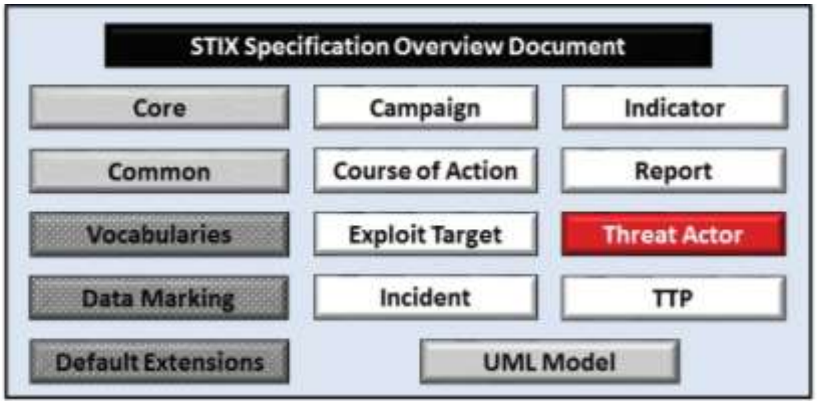

Figure 1-1. STIX[TM] Language v1.2.1 specification documents

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: `AssociatedCampaignsType`, `stixCommon:StatementType`

  Note that all high level concepts have a corresponding UML object. For example, the Threat Actor high level concept is associated with a UML class named, `ThreatActorType`.

- 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 Threat Actor 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 Threat Actor 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 a 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 Threat Actor specification are illustrated via exemplars in **Figure 1-2**.

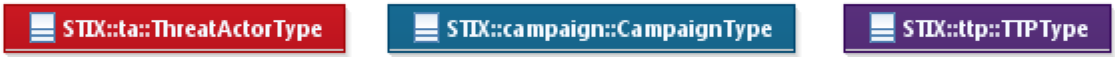

Figure 1-2. Data model color coding

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 Threat Actor 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 <u>verb</u> Y.” For example, in the specification for the STIX Campaign, we write, “The `id` property <u>specifies</u> a globally unique identifier for the Campaign 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**
<u>captures</u>	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 <u>captured</u> include identitifying characteristics, time-related attributes, and a list of the tools used to collect the information. The `Description` property <u>captures</u> 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.
	Example: 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

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

2.1 Threat Actor-Related Component Data Models

As will be explicitly detailed in Section **3**, a STIX Threat Actor leverages two other core STIX constructs, namely Campaign and TTP (as indicated by the outward-oriented arrows). **Figure 2-1** illustrates the relationship between the Threat Actor and the other core constructs. As stated in Section **1.1**, each of these components is defined in a separate specification document.

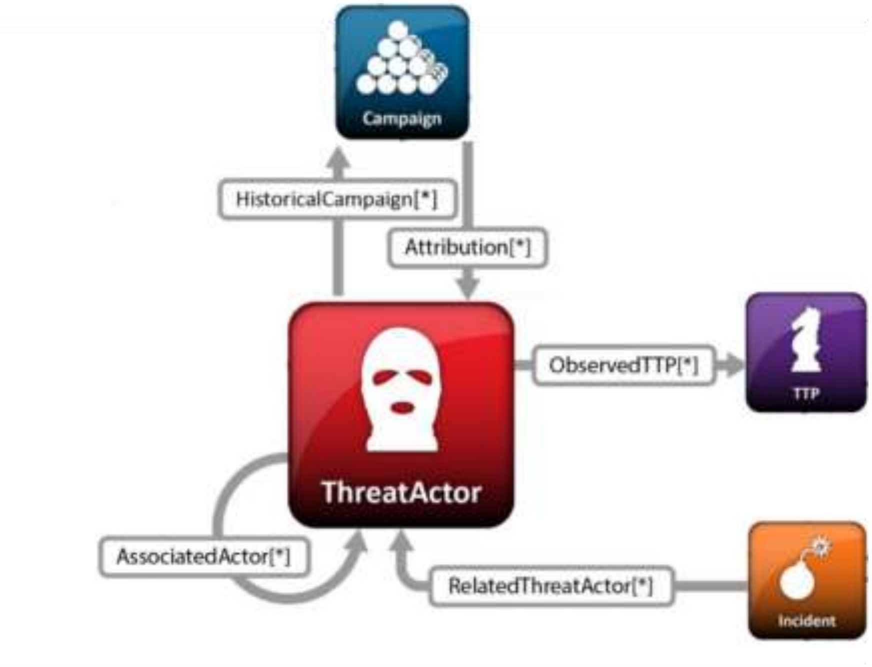

Figure 2-1. High level view of the Threat Actor data model

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

- **Campaign**

  A STIX Campaign represents a set of TTPs, Incidents, or Threat Actors that together express a common intent or desired effect. For example, an adversary using a particular set of TTPs (malware and tools) to target an industry sector with a specific intent may constitute a Campaign. In the STIX data model, a Campaign represents both that intent itself and, perhaps more importantly, acts as a meta-construct to capture the associated TTPs, incidents, and Threat Actors that are part of that Campaign. Please see *STIX Version 1.2.1 Part 8: Campaign* for details.

The Threat Actor data model references the Campaign data model as a means to identify Campaigns thought to be related to the Threat Actor.

- **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.

  The Threat Actor data model references the TTP data model as a means to identify sets of specific TTPs asserted to be leveraged by a Threat Actor (or in some way related to a Threat Actor).

- **Threat Actor**

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

3 STIX[TM] Threat Actor Data Model

The primary class of the STIX Threat Actor package is the `ThreatActorType` class, which characterizes a cyber threat actor including their identity, sophistication, presumed intent, historically observed behavior (TTPs), and campaigns or other threat actors they are believed to be associated with. Similar to the primary classes of all of the component data models in STIX, the `ThreatActorType` class extends a base class defined in the STIX Common data model; more specifically, it extends the `ThreatActorBaseType` base class, which provides the essential identifier (`id`) and identifier reference (`idref`) properties.

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

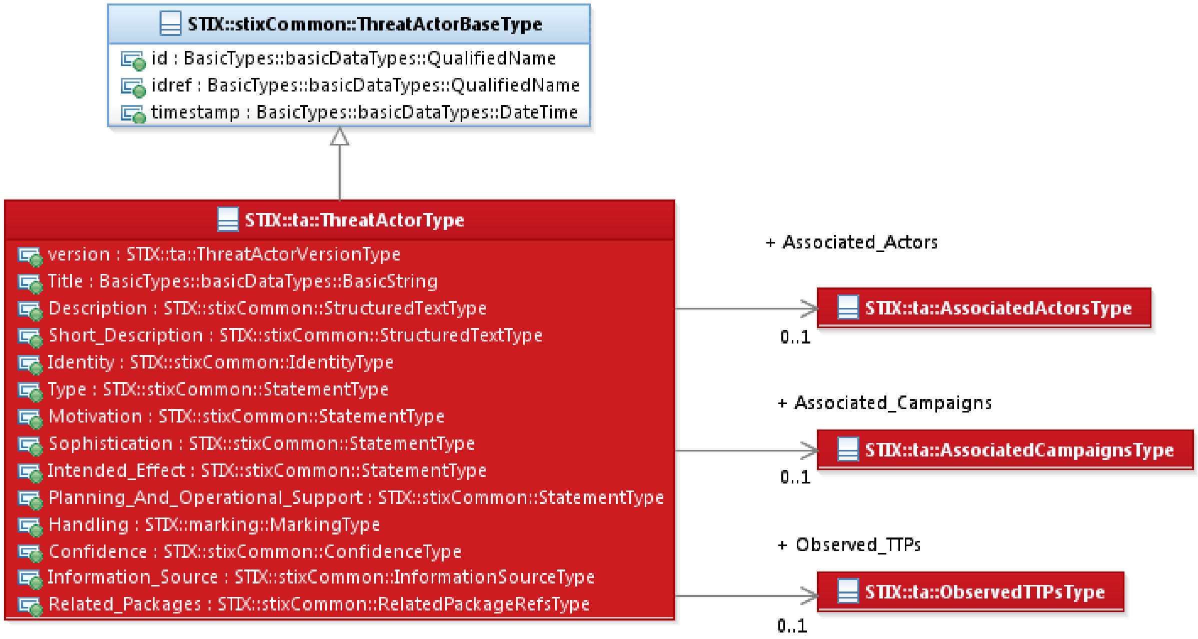
STIX::stixCommon::ThreatActorBaseType
id : BasicTypes::basicDataTypes::QualifiedName
idref : BasicTypes::basicDataTypes::QualifiedName
timestamp : BasicTypes::basicDataTypes::DateTime
STIX::ta::ThreatActorType
version : STIX::ta::ThreatActorVersionType
Title : BasicTypes::basicDataTypes::BasicString
Description : STIX::stixCommon::StructuredTextType
Short_Description : STIX::stixCommon::StructuredTextType
Identity : STIX::stixCommon::IdentityType
Type : STIX::stixCommon::StatementType
Motivation : STIX::stixCommon::StatementType
Sophistication : STIX::stixCommon::StatementType
Intended_Effect : STIX::stixCommon::StatementType
Planning_And_Operational_Support : STIX::stixCommon::StatementType
Handling : STIX::marking::MarkingType
Confidence : STIX::stixCommon::ConfidenceType
Information_Source : STIX::stixCommon::InformationSourceType
Related_Packages : STIX::stixCommon::RelatedPackageRefsType
+ Associated_Actors
STIX::ta::AssociatedActorsType
0..1
+ Associated_Campaigns
STIX::ta::AssociatedCampaignsType
0..1
+ Observed_TTPs
STIX::ta::ObservedTTPsType
0..1

Figure 3-1. UML diagram of the `ThreatActorType` 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**.*

All classes defined in the Threat Actor data model are described in detail in Section **3.1** through Section **3.4**. Details are not provided for classes defined in non-Threat Actor data models; instead, the reader is referred to the corresponding data model specification as indicated by the package prefix specified in the Type column of the table.

Table 3-1. Properties of the `ThreatActorType` *class*

Name	Type	Multiplicity	Description
version	`ThreatActorVersionType`	0..1	The `version` property specifies the version number of the STIX Threat Actor data model for STIX v1.2.1 used to capture the information associated with the Threat Actor.

Title	`basicDataTypes:`
	`BasicString`
		0..1	The `Title` property captures a title for the Threat Actor and reflects what the content producer thinks the Threat Actor as a whole should be called. The `Title` property is typically used by humans to reference a particular Threat Actor; however, it is not suggested for correlation.

Description	`stixCommon:`
	`StructuredTextType`
		0..*	The `Description` property captures a textual description of the Threat Actor. 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 Threat Actor. This property is secondary and should only be used if the `Description` property is already populated and another, shorter description is available.

Identity	`stixCommon:`
	`IdentityType`
		0..1	The `Identity` property characterizes the identity of this Threat Actor. For situations calling for more than a simple name, the underlying class may be extended using a more complete structure such as the `CIQIdentity3.0InstanceType` subclass as defined in *STIX Version 1.2.1 Part 12: Default Extensions*.

Type	`stixCommon:`
	`StatementType`
		0..*	The `Type` property characterizes the type of this Threat Actor, which includes a `Value` property that specifies the particular type of the Threat Actor. Examples of potential types include *black hat hacker, insider threat, and disgruntled customer* (these specific values are only provided to help explain the `Value` property: they are neither recommended values nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary value or may constrain the set of possible types 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 the `Value` property is *'ThreatActorTypeVocab-1.0'* (which is different than the default vocabulary provided for the `StatementType` class).

Motivation	`stixCommon:`
	`StatementType`
		0..*	The `Motivation` property characterizes the motivation of this Threat Actor, which includes a `Value` property that specifies the type of motivation, such as *ego, religious and anti-establishment* (these specific types are only provided to help explain the `Value` property: they are neither recommended types nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary types or may constrain the set of possible types 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 the `Value` property is '*MotivationVocab-1.1'* (which is different than the default vocabulary provided for the `StatementType` class).

Sophistication	`stixCommon:`
	`StatementType`
		0..*	The `Sophistication` property characterizes the sophistication of this Threat Actor, which includes a `Value` property that specifies the level of sophistication. Examples of potential levels include

			innovator, expert, and novice (these specific levels are only provided to help explain the `Value` property: they are neither recommended levels nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary values or may constrain the set of possible values by referencing an externally-defined vocabularyor leveraging a formally defined vocabulary extending from the `stixCommon:ControlledVocabularyStringType` class. The default vocabulary class for use in the `Value` property is '*ThreatActorSophisticationVocab-1.0'* (which is different than the default vocabulary provided for the `StatementType` class).

Intended_Effect	`stixCommon:`
	`StatementType`
		0..1	The `Intended_Effect` property characterizes the suspected intended effect of the Threat Actor, which includes a `Value` property that specifies the type of the effect. Examples of potential types include *theft*, *disruption*, and *unauthorized access* (these specific types are only provided to help explain the `Value` property: they are neither recommended types nor necessarily part of any existing vocabulary). The content creator may choose any arbitrary type or may constrain the set of possible types by referencing an externally-defined vocabularyor leveraging a formally defined vocabulary extending from the `stixCommon:ControlledVocabularyStringType` class. The STIX default vocabulary class for use in the `Value` property is '*IntendedEffectVocab-1.0*' (which is different than the default vocabulary provided for the `StatementType` class).

Planning_And_Operational_Support	`stixCommon:`
	`StatementType`
		0..*	The `Planning_And_Operational_Support` property characterizes suspected planning and operational support available to this Threat Actor, which includes a `Value` property that specifies one type of support, such as *financial, hiring* and *selecting targets* (these specific types are only provided to help explain the `Value` property: they are neither recommended types nor necessarily part of any

			existing vocabulary). The content creator may choose any arbitrary values 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 type for use in the `Value` property is '*PlanningAndOperationalSupportVocab-1.0'* (which is different than the default vocabulary provided for the `StatementType` class).
Observed_TTPs	`ObservedTTPsType`	0..1	The `Observed_TTPs` property specifies a set of one or more TTPs asserted as observed to be leveraged by the Threat Actor (or in some way related to a Threat Actor).
Associated_Campaigns	`AssociatedCampaignsType`	0..1	The `Associated_Campaigns` property specifies a set of one or more Campaigns asserted to be related to the Threat Actor.
Associated_Actors	`AssociatedActorsType`	0..1	The `Associated_Actors` property specifies a set of one or more other Threat Actors asserted to be related to this Threat Actor.
Handling	`marking:MarkingType`	0..1	The `Handling` property specifies the appropriate data handling markings for the properties of this Threat Actor. The marking scope is limited to the Threat Actor and the content is contains. Note that data handling markings can also be specified at a higher level.

Confidence	`stixCommon:`
	`ConfidenceType`
		0..1	The `Confidence` property characterizes the level of confidence in the accuracy of the collection of information captured for the Threat Actor.

Information_Source	`stixCommon:`
	`InformationSourceType`
		0..1	The `Information_Source` property characterizes the source of the Threat Actor information. Examples of details captured include identitifying characteristics, time-related attributes, and a list of tools used to collect the information.

Related_Packages	stixCommon: RelatedPackageRefsType	0..1	The Related_Packages property specifies a set of one or more STIX Packages that are related to the Threat Actor. 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.

3.1 ThreatActorVersionType Enumeration

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

Table 3-2. Literals of the ThreatActorVersionType enumeration

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

3.2 ObservedTTPsType Class

The ObservedTTPsType class specifies a set of one or more TTPs asserted to be leveraged by the Threat Actor (or in some way related to a Threat Actor). 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 ObservedTTPsType class is shown in **Figure 3-2**.

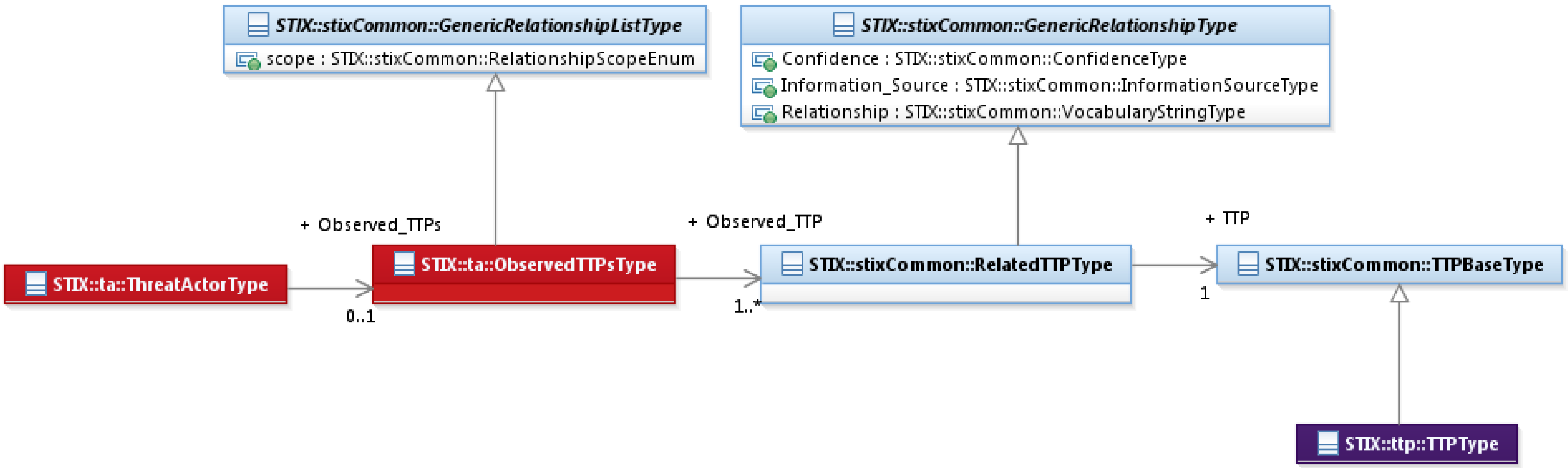
STIX::stixCommon::GenericRelationshipListType
scope : STIX::stixCommon::RelationshipScopeEnum
STIX::stixCommon::GenericRelationshipType
Confidence : STIX::stixCommon::ConfidenceType
Information_Source : STIX::stixCommon::InformationSourceType
Relationship : STIX::stixCommon::VocabularyStringType
+ Observed_TTPs
+ Observed_TTP
+ TTP
STIX::ta::ThreatActorType
STIX::ta::ObservedTTPsType
STIX::stixCommon::RelatedTTPType
STIX::stixCommon::TTPBaseType
0..1
1..*
1
STIX::ttp::TTPType

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

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

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

Name	Type	Multiplicity	Description
Observed_TTP	`stixCommon:RelatedTTPType`	1..*	The `Observed_TTP` property specifies a TTP asserted as observed to be leveraged by the Threat Actor (or in some way related to a Threat Actor) and characterizes the relationship between the Threat Actor and the TTP by capturing information such as the level of confidence that the Threat Actor and the TTP are related, the source of the relationship information, and the type of relationship.

3.3 AssociatedActorsType Class

The `AssociatedActionsType` class specifies one or more other Threat Actors asserted to be related to this Threat Actor 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 `AssociatedActorsType` class is shown in **Figure 3-3**.

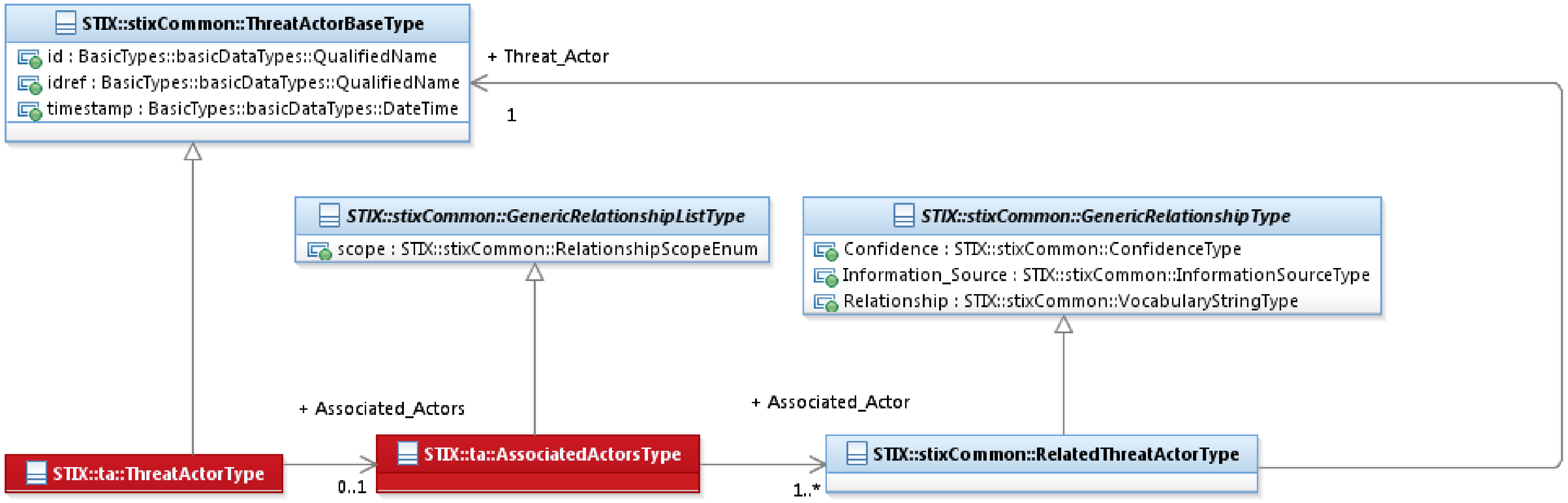
STIX::stixCommon::ThreatActorBaseType
id : BasicTypes::basicDataTypes::QualifiedName
idref : BasicTypes::basicDataTypes::QualifiedName
timestamp : BasicTypes::basicDataTypes::DateTime
+ Threat_Actor
1
STIX::stixCommon::GenericRelationshipListType
scope : STIX::stixCommon::RelationshipScopeEnum
STIX::stixCommon::GenericRelationshipType
Confidence : STIX::stixCommon::ConfidenceType
Information_Source : STIX::stixCommon::InformationSourceType
Relationship : STIX::stixCommon::VocabularyStringType
+ Associated_Actors
+ Associated_Actor
STIX::ta::ThreatActorType
0..1
STIX::ta::AssociatedActorsType
1..*
STIX::stixCommon::RelatedThreatActorType

Figure 3-3. UML diagram of the `AssociatedActorsType` class

Table 3-3 shows the properties of the `AssociatedActorsType` specialization and is associated with the UML diagram given in **Figure 3-3**.

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

Name	Type	Multiplicity	Description

Associated_Actor	`stixCommon:`
	`RelatedThreatActorType`
		1..*	The `Associated_Actor` property specifies another Threat Actor asserted to be associated with this Threat Actor and characterizes the relationship between the Threat Actors by capturing information such as the level of confidence that the Threat Actors are related, the source of the relationship information, and type of the relationship. A relationship between Threat Actors may represent assertions of general associativity or different versions of the same Threat Actor.

3.4 AssociatedCampaignsType Class

The `AssociatedCampaignsType` class specifies a set of one or more of the campaigns asserted to be associated with this Threat Actor. 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 `AssociatedCampaignsType` class is shown in **Figure 3-4**.

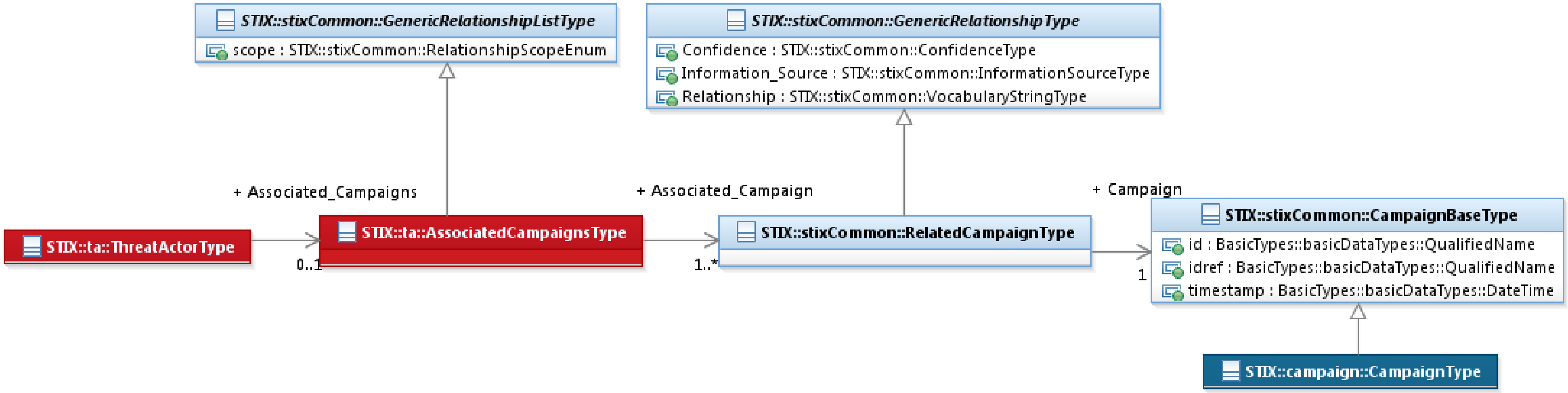
STIX::stixCommon::GenericRelationshipListType
scope : STIX::stixCommon::RelationshipScopeEnum
STIX::stixCommon::GenericRelationshipType
Confidence : STIX::stixCommon::ConfidenceType
Information_Source : STIX::stixCommon::InformationSourceType
Relationship : STIX::stixCommon::VocabularyStringType
+ Associated_Campaigns
+ Associated_Campaign
+ Campaign
STIX::stixCommon::CampaignBaseType
id : BasicTypes::basicDataTypes::QualifiedName
idref : BasicTypes::basicDataTypes::QualifiedName
timestamp : BasicTypes::basicDataTypes::DateTime
STIX::ta::ThreatActorType
0..1
STIX::ta::AssociatedCampaignsType
1..*
STIX::stixCommon::RelatedCampaignType
1
STIX::campaign::CampaignType

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

Table 3-5 shows the properties of the `AssociatedCampaignsType` specialization and is associated with the UML diagram given in **Figure 3-4**.

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

Name	Type	Multiplicity	Description

Associated_Campaign	`stixCommon:`
	`RelatedThreatCampaignType`
		1..*	The `Associated_Campaign` property specifies a Campaign asserted to be associated with this Threat Actor and characterizes the relationship between the Campaign and the Threat Actor by capturing information such as the level of confidence that the Campaign and the Threat Actor are related, the source of the relationship information, and the type of relationship.

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.