STIX Version 1.2.1 Part 7: Threat Actor

Working Draft 01

01 September 2015

Technical Committee:

[OASIS Cyber Threat Intelligence (CTI) TC](https://www.oasis-open.org/committees/cti/)

Chair:

Richard Struse ([Richard.Struse@HQ.DHS.GOV](mailto:Richard.Struse@HQ.DHS.GOV)), [DHS Office of Cybersecurity and Communications (CS&C)](http://www.dhs.gov/office-cybersecurity-and-communications)

Editors:

Sean Barnum ([sbarnum@mitre.org](mailto:sbarnum@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Desiree Beck ([dbeck@mitre.org](mailto:dbeck@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Aharon Chernin ([achernin@soltra.com](mailto:achernin@soltra.com)), [Soltra](http://www.soltra.com/)

Rich Piazza ([rpiazza@mitre.org](mailto:rpiazza@mitre.org)), [MITRE Corporation](http://www.mitre.org/)

Additional artifacts:

This prose specification is one component of a Work Product which consists of:

* *STIX Version 1.2.1 Part 1: Overview*. [URI – added during publication]
* *STIX Version 1.2.1 Part 2: Common*. [URI]
* *STIX Version 1.2.1 Part 3: Core*. [URI]
* *STIX Version 1.2.1 Part 4: Indicator*. [URI]
* *STIX Version 1.2.1 Part 5: TTP*. [URI]
* *STIX Version 1.2.1 Part 6: Incident*. [URI]
* *STIX Version 1.2.1 Part 7: Threat Actor*. (this document)
* *STIX Version 1.2.1 Part 8: Campaign*. [URI]
* *STIX Version 1.2.1 Part 9: Course of Action*. [URI]
* *STIX Version 1.2.1 Part 10: Exploit Target*. [URI]
* *STIX Version 1.2.1 Part 11: Report*. [URI]
* *STIX Version 1.2.1 Part 12: Extensions*. [URI]
* *STIX Version 1.2.1 Part 13: Data Marking*. [URI]
* *STIX Version 1.2.1 Part 14: Vocabularies*. [URI]
* *STIX Version 1.2.1 Part 15: UML Model*. [URI]

Related work:

This specification is related to:

* *CybOX Version 2.1.1 (placeholder)*

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 [Working Draft](https://www.oasis-open.org/policies-guidelines/tc-process#dWorkingDraft) (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or [approved](https://www.oasis-open.org/policies-guidelines/tc-process#committeeDraft) as a Committee Draft (Committee Specification Draft or a Committee Note Draft). The OASIS document [Approval Process](https://www.oasis-open.org/policies-guidelines/tc-process#standApprovProcess) begins officially with a TC vote to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Draft.

URI patterns:

Initial publication URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/csd01/part7-threat-actor/stix-v1.2.1-csd01-part7-threat-actor.docx

Permanent “Latest version” URI:  
http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part7-threat-actor.docx

(Managed by OASIS TC Administration; please don’t modify.)

Copyright © OASIS Open 2015. All Rights Reserved.

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](https://www.oasis-open.org/policies-guidelines/ipr) 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.

Table of Contents

[1 Introduction 4](#_Toc428958982)

[1.1 STIX Specification Documents 4](#_Toc428958983)

[1.2 Document Conventions 5](#_Toc428958984)

[1.2.1 Fonts 5](#_Toc428958985)

[1.2.2 UML Package References 5](#_Toc428958986)

[1.2.3 UML Diagrams 5](#_Toc428958987)

[1.2.3.1 Class Properties 6](#_Toc428958988)

[1.2.3.2 Diagram Icons and Arrow Types 6](#_Toc428958989)

[1.2.3.3 Color Coding 7](#_Toc428958990)

[1.2.4 Property Table Notation 7](#_Toc428958991)

[1.2.5 Property and Class Descriptions 7](#_Toc428958992)

[1.3 Terminology 8](#_Toc428958993)

[1.4 Normative References 8](#_Toc428958994)

[1.5 Non-Normative References 8](#_Toc428958995)

[2 Background 9](#_Toc428958996)

[2.1 Threat Actor-Related Component Data Models 9](#_Toc428958997)

[3 STIX Threat Actor Data Model 11](#_Toc428958998)

[3.1 ThreatActorVersionType Enumeration 16](#_Toc428958999)

[3.2 ObservedTTPsType Class 16](#_Toc428959000)

[3.3 AssociatedActorsType Class 18](#_Toc428959001)

[3.4 AssociatedCampaignsType Class 19](#_Toc428959002)

[4 # Conformance 21](#_Toc428959003)

[Appendix A. Acknowledgments 22](#_Toc428959004)

[Appendix B. Revision History 23](#_Toc428959005)

# Introduction

[All text is normative unless otherwise labeled]

The Structured Threat Information Expression (STIX) framework defines nine top-level component data models: Observable[[1]](#endnote-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 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 Threat Actor data model specification details in Section **3** and conformance information in Section **4**.

## STIX 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*](#AdditionalArtifacts) 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*](#AdditionalArtifacts)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](#AdditionalArtifacts) 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 (default vocabularies, data marking, and extensions), and the color white indicates the component data models. The Observable component data model is shown as an oval shape to indicate that it is defined as a CybOX specification (see [*STIX Version 1.2.1 Part 1: Overview*](#AdditionalArtifacts)for details). 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*](#AdditionalArtifacts).

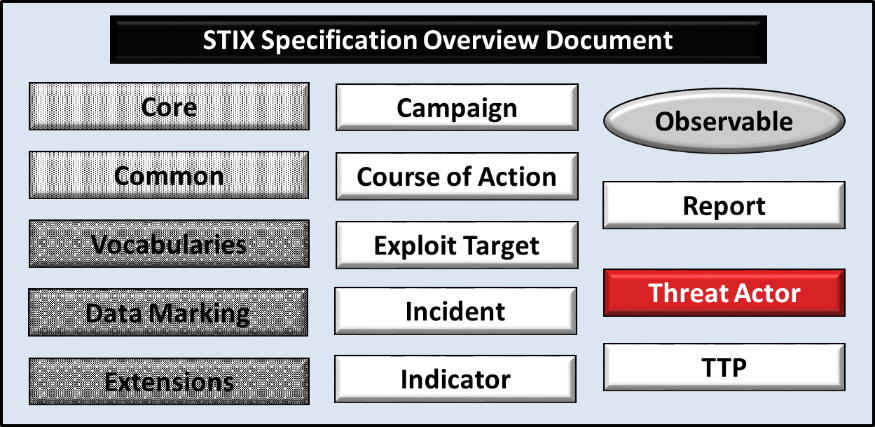


Figure ‑. STIX Language v1.2.1 specification documents

## Document Conventions

The following conventions are used in this document.

### 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*](#AdditionalArtifacts).

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 (withsingle 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.*

### 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*](#AdditionalArtifacts) 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.

### 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.

#### 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.

#### 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 ‑. UML diagram icons

|  |  |
| --- | --- |
| **Icon** | **Description** |
| cid:image003.png@01CFC8F2.37AD2A50 | 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. |

#### 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 ‑. Data model color coding

### 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\_CallA, CodeB) and single logic expression in the Multiplicity column.  For example, if there is a choice of property API\_CallA and CodeB, 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.

### Property and Class Descriptions

Each class and property defined in STIX is described using the format, “The X property verbY.” For example, in the specification for the STIX Campaign, we write, “The id property specifies 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** |
| 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. |
|  | *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. |

## 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].

## 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>.

## Non-Normative References

# 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**.

## 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.

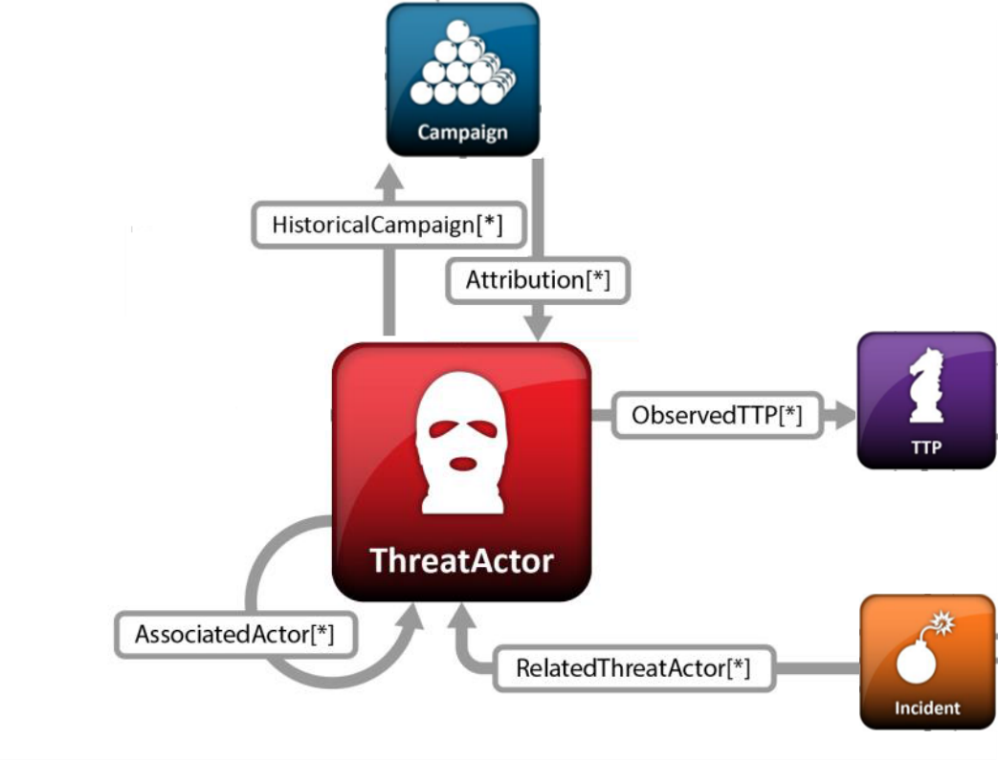


Figure ‑. High level view of the Threat Actor data model

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](http://stixproject.github.io/data-model/1.1.1/campaign/CampaignType) 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*](#AdditionalArtifacts) 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*](#AdditionalArtifacts) 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.

# 

# STIX 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**.



Figure ‑. 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 ‑. 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: Extensions*](#AdditionalArtifacts). |
| **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. |

## 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 ‑. Literals of the ThreatActorVersionType enumeration

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

## 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**.



Figure ‑. 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 ‑. 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. |

## 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**.



Figure ‑. 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 ‑. 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. |

## 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**.



Figure ‑. 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 ‑. 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. |

# # Conformance

The last numbered section in the specification must be the Conformance section. Conformance Statements/Clauses go here. [Remove # marker]

1. Acknowledgments

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

Participants:

[Participant Name, Affiliation | Individual Member]

[Participant Name, Affiliation | Individual Member]

1. Revision History

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

1. The CybOX Observable data model is actually defined in the [CybOX Language](#RelatedWork), not in STIX. [↑](#endnote-ref-1)