STIX 2.0 Specification

Version 2.0-draft-2

Document Table of Contents

[​1.​ Technical Committee](#h.gxv050qkafk2)

[​2.​ Chair](#h.3jgkzl5fxn64)

[​3.​ Editors](#h.ju54f7z963iq)

[​4.​ Acknowledgments](#h.fyqa1bmkq837)

[​5.​ Abstract](#h.3gzz4i55o6zt)

[​6.​ Introduction](#h.sco4ewvqn4ax)

[​6.1.​ Overview](#h.gd1y9dfvz9nh)

[​6.1.1.​ Graph-Based Model](#h.pnsw76i2hf9j)

[​6.1.2.​ STIX Domain Objects](#h.1j0vun2r7rgb)

[​6.1.3.​ STIX Relationships](#h.bsg4a0ll0fd6)

[​6.1.4.​ Vocabularies](#h.vbsdt43uxrv0)

[​6.1.5.​ Serialization](#h.vj2dopx186bb)

[​6.1.6.​ Sharing STIX](#h.65z4ktwggddy)

[​6.2.​ Requirements](#h.jdoduhys9iz)

[​6.3.​ Document Structure](#h.dixoncb7i3ou)

[​6.4.​ Conventions](#h.38mgz8dpg8z3)

[​6.4.1.​ Naming Conventions](#h.pc5gnp8dofa)

[​6.4.2.​ Reserved Property Names](#h.il7pzshahd9b)

[​6.4.3.​ Font Colors and Style](#h.601hcnud1rpt)

[​7.​ Common Data Types](#h.yeo5yj6uksa9)

[​7.1.​ Boolean](#h.m3ape5gjvhgw)

[​7.1.1.​ Examples](#h.rtdnfjbv9hx5)

[​7.2.​ CybOX Container](#h.72bcfr3t79jx)

[​7.3.​ External Reference](#h.cez46v5quobo)

[​7.3.1.​ Properties](#h.bajcvqteiard)

[​7.3.2.​ Requirements](#h.qe0c6m84wete)

[​7.3.3.​ Examples](#h.4d7sk86v223x)

[​7.4.​ Identifier](#h.ko24ggw4eq0q)

[​7.4.1.​ Examples](#h.x0g8rwb1crez)

[​7.5.​ Kill Chain Phase](#h.i4tjv75ce50h)

[​7.5.1.​ Examples](#h.toy67t7ucwu0)

[​7.6.​ List](#h.9w329aiwpu1y)

[​7.7.​ MAEC™](#h.24yz7mq2rzyk)

[​7.8.​ Number](#h.uxyhzmv0vpyc)

[​7.8.1.​ Examples](#h.ovzsxfpu1kvg)

[​7.9.​ String](#h.i51xum143796)

[​7.9.1.​ Examples](#h.nl2yaiphhghz)

[​7.10.​ Timestamp](#h.de8ah59mobqf)

[​7.10.1.​ Requirements](#h.i7rrdtzgta8d)

[​7.10.2.​ Examples](#h.nwcyih3nuzgg)

[​7.11.​ Timestamp Precision](#h.sp8ake5xbk8j)

[​7.11.1.​ Requirements](#h.p51olxypg7ym)

[​7.11.2.​ Examples](#h.h7ok55po12eb)

[​7.12.​ Open Vocabulary](#h.karbmftow040)

[​7.12.1.​ Examples](#h.178gr6yu91g7)

[​8.​ STIX Objects](#h.sfz7m3er434p)

[​8.1.​ Common Properties](#h.xzbicbtscatx)

[​8.2.​ IDs and References](#h.h02ac9vlmabi)

[​8.3.​ Object Creator](#h.a35wk7dyf9q5)

[​8.4.​ Versioning](#h.rye5q2hkacu)

[​8.4.1.​ Versioning Timestamps](#h.wc24wqtv2k5l)

[​8.4.2.​ New Version or New Object?](#h.vy23e1urdh0)

[​8.4.3.​ Examples](#h.pt0fxyc2czwe)

[​8.5.​ Common Relationships](#h.f3dx2rhc3vl)

[​8.6.​ Reserved Properties](#h.p54wxsxz1lsm)

[​9.​ Data Handling in STIX](#h.6zghk3evovcs)

[​9.1.​ Marking Definition](#h.k5fndj2c7c1k)

[​9.1.1.​ Properties](#h.8grkny2a82l3)

[​9.1.2.​ Relationships](#h.h8809y5grg3)

[​9.1.3.​ TLP Marking Object Type](#h.yd3ar14ekwrs)

[​9.1.4.​ Statement Marking Object Type](#h.iwkptagz813n)

[​9.1.5.​ Examples](#h.fxaluk5wypck)

[​9.2.​ Object Markings](#h.bnienmcktc0n)

[​9.2.1.​ Examples](#h.urx7nishhrsz)

[​9.3.​ Granular Markings](#h.robezi5egfdr)

[​9.3.1.​ Granular Marking Type](#h.l6edgya0tyjq)

[​9.3.1.1.​ Selector Syntax](#h.5jigd52xda5r)

[​9.3.2.​ Example](#h.oxqlhgi8obyh)

[​10.​ STIX Domain Objects](#h.rvtdrdkf1jdv)

[​11.​ Relationship Objects](#h.ceolg84lu8th)

[​12.​ Metadata Objects](#h.dnm4ez5y24uh)

[​13.​ Vocabularies](#h.iit7tolczlxv)

[​13.1.​ Attack Motivation](#h.dvmbnm1zpjbt)

[​13.2.​ Attack Objective](#h.ueut8mnjgrc1)

[​13.3.​ Attack Resource Level](#h.nlxwwbc73m4e)

[​13.4.​ Attack Sophistication Level](#h.be1dktvcmyu)

[​13.5.​ Course of Action Label](#h.xblzfl7rn1b7)

[​13.6.​ Identity Classification](#h.e944wydbd83w)

[​13.7.​ Incident Label](#h.cvhfwe3t9vuo)

[​13.8.​ Indicator Label](#h.a50wvo4z81ef)

[​13.9.​ Industry Sector](#h.oogrswk3onck)

[​13.10.​ Malware Label](#h.8cyb6e9yqzwr)

[​13.11.​ Pattern Language](#h.ny5k78eru8kd)

[​13.12.​ Report Label](#h.bijmxibgkk5m)

[​13.13.​ Threat Actor Label](#h.tqbl8z36yoir)

[​13.14.​ Threat Actor Role](#h.u6befh8d18r)

[​13.15.​ Tool Label](#h.cozm95emj8qk)

[​14.​ Customizing STIX](#h.4ne27rjj6udo)

[​14.1.​ Custom Properties](#h.8072zpptza86)

[​14.1.1.​ Requirements](#h.3a2x3jdr23tq)

[​14.1.2.​ Examples](#h.jcxkyx38l8bz)

[​14.2.​ Custom Objects](#h.7f3c4jgkyhl3)

[​14.2.1.​ Requirements](#h.u7ks5xud8vj0)

[​14.2.2.​ Examples](#h.8gs7jkwodtw6)

# ​1.​ Technical Committee

OASIS Cyber Threat Intelligence (CTI) TC

# ​2.​ Chair

Richard Struse (Richard.Struse@hq.dhs.gov), DHS Office of Cybersecurity and Communications (CS&C)

# ​3.​ Editors

Bret Jordan ([bret.jordan@bluecoat.com](mailto:bret.jordan@bluecoat.com)), Blue Coat Systems, Inc.

John Wunder ([jwunder@mitre.org](mailto:jwunder@mitre.org)), MITRE Corporation

# ​4.​ Acknowledgments

**STIX Subcommittee Chairs:**

John Wunder ([jwunder@mitre.org](mailto:jwunder@mitre.org)), MITRE Corporation

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

**Special Thanks:**

The following individuals made substantial contributions to this specification in the form of normative text and proofing and their contributions are gratefully acknowledged:

* Bret Jordan, Blue Coat Systems, Inc.
* Terry MacDonald, Cosive
* Jane Ginn, Cyber Threat Intelligence Network, Inc. (CTIN)
* Richard Struse, DHS Office of Cybersecurity and Communications
* Jason Keirstead, IBM
* Tim Casey, Intel
* Allan Thomson, LookingGlass Cyber
* John Wunder, MITRE Corporation
* Richard Piazza, MITRE Corporation
* John-Mark Gurney, New Context Services, Inc.
* Iain Brown, United Kingdom Cabinet Office

**Contributors:**

The following individuals were members of the OASIS CTI Technical Committee during the creation of this specification and their contributions are gratefully acknowledged:

<todo, make sure this is up to date>

* Dean Thompson, Australia and New Zealand Bank
* Alexander Foley, Bank of America
* Bret Jordan, Blue Coat Systems, Inc.
* Sarah Kelley, Center for Internet Security (CIS)
* Alexandre Dulaunoy, CIRCL
* Ted Bedwell, Cisco Systems
* Craig Brozefsky, Cisco Systems
* Jyoti Verma, Cisco Systems
* Jane Ginn, Cyber Threat Intelligence Network, Inc. (CTIN)
* Richard Struse, DHS Office of Cybersecurity and Communications
* Marlon Taylor, DHS Office of Cybersecurity and Communications (CS&C)
* Gordan Hundley, DTCC
* Wouter Bolsterlee, EclecticIQ
* Joep Gommers, EclecticIQ
* Sergey Rolzunov, EclecticIQ
* Rutger Prins, EclecticIQ
* Andrei Sirghi, EclecticIQ
* Raymon van der Velde, EclecticIQ
* Ravi Sharda, EMC
* David Eilken, FS-ISAC
* Paul Patrick, FireEye, Inc.
* Sarah Brown, Fox-IT
* Ryusuke Masuoka, Fujitsu Limited
* Daisuke Murabayashi, Fujitsu Limited
* Eric Burger, Georgetown University
* Masato Terada, Hitachi, Ltd.
* Jason Keirstead, IBM
* Paul Martini, iboss, Inc.
* Jerome Athias, Individual
* Sanjiv Kalkar, Individual
* Terry MacDonald, Cosive
* Alex Pinto, Individual
* Tim Casey, Intel Corporation
* Julie Modlin, Johns Hopkins University, Applied Physics Laboratory
* Mark Moss, Johns Hopkins University, Applied Physics Laboratory
* Pamela Smith, Johns Hopkins University, Applied Physics Laboratory
* Allan Thomson, LookingGlass Cyber
* Greg Back, MITRE Corporation
* Jonathan Baker, MITRE Corporation
* Sean Barnum, MITRE Corporation
* Nicole Gong, MITRE Corporation
* Ivan Kirillov, MITRE Corporation
* Richard Piazza, MITRE Corporation
* Jon Salwen, MITRE Corporation
* Emmanuelle Vargas-Gonzalez, MITRE Corporation
* Bryan Worrell, 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.
* Robin Cover, OASIS
* Chet Ensign, OASIS
* Cory Casanave, Object Management Group
* John Tolbert, Queralt, Inc.
* Igor Baikalov, Securonix
* Bernd Grobauer, Siemens AG
* John Anderson, Soltra
* Aharon Chernin, Soltra
* Trey Darley, Kingfisher Operations, sprl
* Mark Davidson, Soltra
* Paul Dion, Soltra
* Daniel Dye, Soltra
* Ali Khan, Soltra
* Natalie Suarez, Soltra
* Cedric LeRoux, Splunk, Inc.
* Brian Luger, Splunk, Inc.
* Tom Blauvelt, Symantec Corp.
* Crystal Hayes, The Boeing Company
* Andrew Pendergast, ThreatConnect, Inc.
* Brad Butts, U.S. Bank
* Brian Fay, U.S. Bank
* Mona Magathan, U.S. Bank
* Yevgen Sautin, U.S. Bank
* Iain Brown, United Kingdom Cabinet Office
* Adam Cooper, United Kingdom Cabinet Office
* Mike McLellan, United Kingdom Cabinet Office
* Chris O’Brien, United Kingdom Cabinet Office
* Howard Staple, United Kingdom Cabinet Office
* Chris Taylor, United Kingdom Cabinet Office
* Laurie Thomson, United Kingdom Cabinet Office
* Julian White, United Kingdom Cabinet Office
* Bethany Yates, United Kingdom Cabinet Office
* Eoghan Casey, US Department of Defense (DoD)
* Gary Katz, US Department of Defense (DoD)
* Jeff Mates, US Department of Defense (DoD)
* Robert Coderre, VeriSign
* Haripriya Gajendran, VeriSign
* Kyle Maxwell, VeriSign
* Eric Osterweil, VeriSign
* Anthony Rutkowski, Yaana Technologies, LLC

# ​5.​ Abstract

TODO: We will add this once the rest of the document is completed.

# ​6.​ Introduction

Structured Threat Information Expression (STIX™) is an international collaborative project to define an information exchange language and serialization for the purpose of specifying, characterizing and capturing cyber threat intelligence (CTI). STIX enables organizations to share cyber threat intelligence with one another in a consistent manner, allowing security communities to better understand what computer-based attacks they are most likely to see, and to respond to those attacks faster and more effectively. STIX is designed to improve many different scenarios, such as collaborative threat analysis, automated threat exchange at scale, automated detection and response and many more.

In response to lessons learned in implementing previous versions, STIX has been significantly redesigned and, as a result, omits some of the objects and fields defined in STIX 1.2.1. The objects chosen for inclusion in STIX 2.0 represent a minimally viable product (MVP) to fulfill basic consumer and producer requirements for cyber threat intelligence sharing. Objects and fields not included in STIX 2.0, but deemed necessary by the community, will be included in future 2.x releases (development of STIX 2.1 will immediately follow the STIX 2.0 release).

## ​6.1.​ Overview

### ​6.1.1.​ Graph-Based Model

The STIX 2.0 language is graph-based, where STIX Domain Objects (SDOs) define the graph nodes and STIX relationships (including STIX Relationship Objects and embedded references) define the edges. This graph-based language conforms to common analysis approaches and allows for flexible, modular, structured, and consistent representations of cyber threat intelligence.

### ​6.1.2.​ STIX Domain Objects

STIX 2.0 defines fourteen STIX Domain Objects (SDOs): Attack Pattern, Campaign, Course of Action, Incident, Indicator, Intrusion Set, Malware, Observed Data, Report, Source, Threat Actor, Tool, Victim Target, and Vulnerability. Those objects each correspond to a concept commonly represented in cyber threat intelligence. Using the relationships, they can then be used as building blocks and composed into broader intelligence pictures.

SDOs are STIX Objects and all share a common set of properties. These properties provide standard capabilities such as versioning, data marking (representing how data can be shared and used), and extensibility.

### ​6.1.3.​ STIX Relationships

There are several different types of relationships in STIX:

* *Factual relationships*, where the relationship is known as a fact, are represented as direct references embedded within objects. An example is the **created\_by\_ref** property, which represents the Source that created a piece of STIX content via its **id**.
* *Intelligence relationships*, where the relationship is an analytical assertion made by the content creator, link SDOs together via a STIX Relationship Object (SRO). For example, an Incident can be linked to a Campaign with an SRO to assert that the Incident is attributed to that Campaign.

There are currently forty-four named intelligence relationships defined by the STIX 2.0 specification and represented using the generic Relationship SRO. If none of those suffice, a content producer can use the generic "related-to" common relationship or create a new custom relationship name, also using the Relationship SRO.

Some intelligence relationships require additional data to fully describe the relationship. These relationships are represented as specific types of SROs. The Sighting SRO, for example, is used to relate SDOs with Observed Data to indicate that those SDOs were sighted. It defines a **count**, **first\_seen**, and **last\_seen** properties to capture extra data about the relationship that are not present on the generic Relationship SRO.

SROs are STIX Objects and use the same common properties as SDOs. This provides standard capabilities such as versioning, data marking (representing how data can be shared and used), and extensibility.

### ​6.1.4.​ Vocabularies

STIX 2.0 defines a set of suggested vocabularies to promote interoperability while still maintaining flexibility. The vocabularies are used within STIX Objects to give consistent context in the information exchange model. Vocabularies in STIX are "open": while the STIX specification defines a set of suggested terms that are commonly agreed on, well-defined, and well-understood, end users have the ability to add custom terms.

### ​6.1.5.​ Serialization

STIX 2.0 is designed to support multiple serializations, however the mandatory-to-implement (MTI) serialization in STIX 2.0 is JSON (TODO REF IETF). This means that all STIX-compatible tools **MUST** support JSON as a serialization and can optionally support additional serialization formats.

As JSON is the mandatory-to-implement serialization, all examples in this document are expressed in JSON.

### ​6.1.6.​ Sharing STIX

STIX 2.0 is designed to be transport-agnostic. The structures and serializations do not require any specific transport mechanism. A companion CTI specification, TAXII (TODO REF), was designed specifically to transport STIX Objects and is the recommended mechanism to do so. If TAXII connectivity is not available or desired, STIX provides the Bundle object to allow for transportation of STIX data over non-TAXII communication mechanisms.

## ​6.2.​ Requirements

The keywords “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “NOT RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 [TODO add reference].  
  
An implementation is not compliant if it fails to satisfy one or more of the **MUST** or **REQUIRED** level requirements. An implementation that satisfies all the **MUST** or **REQUIRED** level and all the **SHOULD** level requirements is said to be “unconditionally compliant”; one that satisfies all the **MUST** level requirements but not all the **SHOULD** level requirements is said to be “conditionally compliant”.

## ​6.3.​ Document Structure

This specification document is structured as follows.

<TODO - CHECK THE ORDER OF THE SECTIONS AND ORGANISE THIS PARA AS APPROPRIATE>

Section <TODO> defines the common types used throughout STIX, and is referenced in other sections of this specification. Information on customizing STIX can be found in Section 4<TODO>, with guidance, requirements and examples of Custom Properties. Section 5<TODO> outlines how STIX Objects are transported or transmitted. Section 6<TODO> defines the properties and behaviors common to all STIX Domain Objects (SDOs).

<to do> FINISH EXPANDING THIS SECTION - Suggest we do this when we have combined all the STIX docs into one. Easier then to get order of sections correct.

## ​6.4.​ Conventions

### ​6.4.1.​ Naming Conventions

All type names, property names and literals are in lowercase. Words in property names are separated with an underscore (\_), while words in type names and string enumerations are separated with a dash (-). All type names, property names, object names, and vocabulary terms are between three and 250 characters long.

### ​6.4.2.​ Reserved Property Names

Reserved property names are marked with a type called RESERVED and a description text of “RESERVED FOR FUTURE USE”. Any property name that is marked as RESERVED **MUST NOT** be used in any implementation.

### ​6.4.3.​ Font Colors and Style

The following color, font and font style conventions are used in this document:

* The Consolas font is used for all type names, property names and literals.
  + type names are in red with a light red background – package
  + property names are in bold style – **created\_at**
  + literals (values) are in green with a green background – IP Watchlist
  + as “named relationship” are string literals, they will also appear in green with a green background – related-to
* In property tables, if the property is being redefined from an inherited value in some way, then the background is dark grey.
* All examples in this document are expressed in the JSON. They are in Consolas 9 pt font, with straight quotes, black text and a light blue background. All examples have a 2 space indentation.

​

# ​7.​ Common Data Types

**TODO Open questions:**

1. Make sure each of the sections are consistent with each other once we're done
2. Need to figure out how to do statements about JSON MTI serialization

This section defines the common types used throughout STIX. These types will be referenced by the “Type” column in other sections. This section defines the names and permitted values of common types that are used in the STIX information model; it does not, however, define the meaning of any fields using these types. These types may be further restricted elsewhere in the document.

|  |  |
| --- | --- |
| **Type** | **Description** |
| boolean | Contains a value of true or false. |
| cybox-container | A container for CybOX content. This type is defined by the [TODO CybOX Reference]. |
| external-reference | Contains a non-STIX identifier or reference to other related external content. |
| identifier | Contains an identifier (ID) for a STIX Object. |
| kill-chain-phase | Contains a reference to a kill chain phase by name. |
| list | Contains an ordered sequence of values. Often, the phrasing “list of type <type>” is used to indicate that all values within the list must conform to a specific type. |
| maec-container | A container for MAEC™ content. This type is defined by [TODO Ref MAEC]. |
| number | Contains a number. |
| open-vocab | Contains a value from a STIX open (open-vocab) or suggested vocabulary. |
| string | Contains text. |
| timestamp | Contains a time value (date and time). |
| timestamp-precision | Contains the level of precision for timestamps. |

​

## ​7.1.​ Boolean

**Type Name:** boolean

A boolean contains a value of either true or false. Properties with this type **MUST** have a value of true or false.

The JSON MTI serialization uses the JSON boolean type <TODO: add reference>, which is a literal (unquoted) true or false.

### ​7.1.1.​ Examples

{

...

"summary": true,

...

}

## ​7.2.​ CybOX Container

**Type Name:** cybox-container

A container for CybOX content, as defined by [TODO Ref CybOX]. The structure is duplicated here simply as a reference; normative usage is defined by the CybOX language.

<todo: When CybOX is finished, copy in>

## ​7.3.​ External Reference

**Type Name:** external-reference

External references are used to describe pointers to information represented outside of STIX. For example, an Incident could use an external reference to indicate an ID for that incident in an external database or a report could use references to represent source material.

### ​7.3.1.​ Properties

|  |  |  |
| --- | --- | --- |
| Property Name | Type | Description |
| **source** (required) | string | The source within which the external-reference is defined (system, registry, organization, etc.). |
| **description** (optional) | string | A human readable description. |
| **url** (optional) | url | A URL reference to an external resource. [TODO: Reference to URL syntax] |
| **external\_id** (optional) | string | An identifier for the external reference content. |

### ​7.3.2.​ Requirements

* At least one of **external\_id**, **url**, and **description** fields **MUST** be present.

### ​7.3.3.​ Examples

An external-reference from the CAPEC™ (TODO add ref) repository

{

...

"external\_references": [

{

"source": "capec",

"external\_id": "CAPEC-550"

}

],

...

}

An external-reference from the CAPEC repository with URL

{

...

"external\_references": [

{

"source": "capec",

"external\_id": "CAPEC-550",

"url": "<http://capec.mitre.org/data/definitions/550.html>"

}

],

...

}

An external-reference to Mandiant’s APT1 report document

{

...

"external\_references": [

{

"source": "Mandiant",

"description": "APT1 report",

"url": "<http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf>"

}

],

...

}

An external-reference to a VERIS [Community Database (VCDB)](http://www.vcdb.org) [TODO:Add ref?] entry

{

...

"external\_references": [

{

"source": "veris",

"external\_id": "0001AA7F-C601-424A-B2B8-BE6C9F5164E7",

"url": "https://github.com/vz-risk/VCDB/blob/master/data/json/0001AA7F-C601-424A-B2B8-BE6C9F5164E7.json"

}

],

...

}

An external-reference to a Jira item

{

...

"external\_references": [

{

"source": "jira",

"external\_id": "TAB-1370",

"url": "<https://issues.oasis-open.org/browse/TAB-1370>"

}

],

...

}

## ​7.4.​ Identifier

**Type Name:** identifier

An identifier universally and uniquely identifies an instance of a STIX Object. Identifiers **MUST** follow the form [object-type]--[UUIDv4], where **[object-type]** is the exact value from the type field of the object being identified or referenced and where the **[UUIDv4]** is an RFC 4122-compliant Version 4 UUID. The UUID **MUST** be generated according to the algorithm(s) defined in RFC 4122, Section 4.4 (Version 4 UUID) [add reference].

### ​7.4.1.​ Examples

{

...

"type": "indicator",

"id": "indicator--e2e1a340-4415-4ba8-9671-f7343fbf0836",

...

}

## ​7.5.​ Kill Chain Phase

**Type Name:** kill-chain-phase

The kill-chain-phase represents a phase in a kill chain. The concept of a kill chain, as applied to information security by Lockheed Martin in its Cyber Kill Chain™ [TODO add reference], is to elaborate the various phases a Threat Actor may undertake in order to compromise a system.

|  |  |  |
| --- | --- | --- |
| Property Name | Type | Description |
| **kill\_chain\_name** (required) | string | The name of the kill chain. The value of this field **SHOULD** be all lowercase (where lowercase is defined by the locality conventions) and **SHOULD** use dashes instead of spaces or underscores. |
| **phase\_name** (required) | string | The name of the phase in the kill chain. The value of this field **SHOULD** be all lowercase (where lowercase is defined by the locality conventions) and **SHOULD** use dashes instead of spaces or underscores. |

### ​7.5.1.​ Examples

{

...

"kill\_chain\_phases": [

{

"kill\_chain\_name": "kill-chain-foo",

"phase\_name": "phase-foo"

}

],

...

}

## ​7.6.​ List

**Type Name:** list

A list contains an ordered sequence of values. When the phrasing “list of type <type>” is used, all values in the list **MUST** be of the specified type. For instance, list of type number means that all values of the list must be of the number type. Upper and lower bounds of the list – the minimum and maximum number of values – may be specified where the list is used. This section does not specify the maximum number of allowed values in a list, however every instance of a list **MUST** have at least one value. Therefore, empty lists are prohibited in STIX and **MUST NOT** be used as a replacement for omitting the property if the list is optional.

The JSON MTI serialization uses the JSON array type [TODO: Add ref?], which is an ordered list of zero or more values.

​7.6.1. Examples

{

...

"observed\_data\_refs": [

"observed-data--b67d30ff-02ac-498a-92f9-32f845f448cf",

"observed-data--c96f4120-2b4b-47c3-b61f-eceaa54bd9c6",

"observed-data--787710c9-1988-4a1b-9761-a2de5e19c62f"

],

...

}

## ​7.7.​ MAEC™

**Type Name:** maec-container

A container for MAEC content, as defined by [TODO Ref MAEC]. The structure is duplicated here simply as a reference; normative usage is defined by the MAEC language.

[TODO: When MAEC 5.0 is finished, copy in].

​

## ​7.8.​ Number

**Type Name:** number

The number type represents any number that can be expressed as a real number (e.g., -10, 0, 10, 10.1, 10.123213). Each use of the number type may specify the following:

* The valid range of values;
* Whether it is limited to integers or not; and
* The maximum number of decimal places.

In the JSON MTI serialization, numbers are represented by the JSON number type [TODO: Add reference].

### ​7.8.1.​ Examples

{

...

"count": 8,

...

}

## ​7.9.​ String

**Type Name:** string

The string data type represents a finite-length string of valid characters from the Unicode coded character set [ISO.10646]. Unicode incorporates ASCII [RFC20] and the characters of many other international character sets.

The JSON MTI serialization uses the JSON string type [TODO: Add reference], which mandates the UTF-8 encoding for supporting Unicode.

### ​7.9.1.​ Examples

{

...

"title": "The Black Vine Cyberespionage Group",

...

}

​

## ​7.10.​ Timestamp

**Type Name:** timestamp

The timestamp type defines how timestamps are represented in STIX. Most discrete timestamps (i.e., not time ranges or relative times) in STIX have a corresponding optional field that indicates the precision of the timestamp, of type timestamp-precision.

In cases where the timestamp is metadata about the STIX construct, such as creation and modification times for STIX Objects, the timestamp field will not have the corresponding precision field. In these cases, the timestamp should be treated as if the precision field is full.

### ​7.10.1.​ Requirements

* The timestamp field **MUST** be a valid RFC 3339-formatted timestamp [TODO add reference] using the format YYYY-MM-DDTHH:mm:ss[.s+]Z where the “s+” represents 1 or more sub-second values.
* The timestamp **MUST** be represented in the UTC timezone and **MUST** use the “Z” designation to indicate this.

### ​7.10.2.​ Examples

A timestamp that does not have a precision field defined would look like:

{

...

"created": "2016-01-20T12:31:12.12345Z",

...

}

## ​7.11.​ Timestamp Precision

**Type Name:** timestamp-precision

A timestamp-precision represents the precision options for a given timestamp.

### ​7.11.1.​ Requirements

* If present, the timestamp-precision field **MUST** have a value of year, month, day, hour, minute, or full.
  + The default value for the precision field is full, so omitting the field is equivalent to explicitly specifying full.
  + A value of full indicates that the value in the timestamp field is precise to the full number of digits in the timestamp value (including any fractional seconds, such as milliseconds or microseconds).
  + A value of minute, hour, day, month, or year indicates that the timestamp value is precise to that as a lower bound (the precision window is the timestamp value plus one unit of the precision value).
    - *For example, if the timestamp value is 2016-04-25T13:00:00Z and the precision value is hour, the time is greater than or equal to 2016-04-25T13:00:00Z and less than 2016-04-25T14:00:00Z.*
  + When specifying a precision other than full, the time portion of the timestamp field **MUST** contain 00 for all fields beyond the specified precision while the date portion **MUST** contain 01 for all fields beyond the specified precision.
    - *For example, if the precision field is* month*, the* timestamp *field must contain 01 for the day field and 00 for the hour, minute, and second fields such as 2016-12-01T00:00:00Z.*
* The timestamp-precision field will always be nested at the same level as the timestamp field.
* The property name for the precision field is **[timestamp\_field\_name]\_precision**.
  + *For example, if the key of the* timestamp *field is* ***valid\_from****, the key of the precision field is* ***valid\_from\_precision****.*

### ​7.11.2.​ Examples

**The following examples have explicitly defined the precision**

A timestamp known only to a year would look like:

{

...

"start": "2016-01-01T00:00:00Z",

"start\_precision": "year",

...

}

A timestamp known only to an hour would look like:

{

...

"end": "2016-01-20T12:00:00Z",

"end\_precision": "hour",

...

}

**The following examples have implicitly defined the precision**

A timestamp known to a second would look like:

{

...

"start": "2016-01-20T12:31:12Z",

...

}

A timestamp known to 5-digit sub-second precision would look like:

{

...

"end": "2016-01-20T12:31:12.12345Z",

...

}

## ​7.12.​ Open Vocabulary

**Type Name:** open-vocab

An open vocabulary is a string field that provides a list of suggested values, without constraining producers from extending those values. The list of suggested values is known as the suggested vocabulary. The value of an open-vocab field **SHOULD** be a value from the suggested vocabulary but **MAY** be any other string value. Values that are not from the value list **SHOULD** be all lowercase (where lowercase is defined by the locality conventions) and **SHOULD** use dashes instead of spaces or underscores.

### ​7.12.1.​ Examples

**Example using value from the suggested vocabulary**

In this example the Indicator **labels** property is an open vocabulary and we are using one of the suggested vocabulary values.

{

...,

"labels": ["malicious-activity"],

...

}

**Example using a custom value**

In this example, for the same Indicator **labels** property, we are not using a value in the suggested vocabulary.

{

...,

"labels": ["pbx-fraud-activity"],

...

}

# ​8.​ STIX Objects

This section outlines the common properties and behavior across all STIX Objects, including STIX Domain Objects, STIX Relationship Objects, and the Marking Definition Object (see Section TODO).

## ​8.1.​ Common Properties

This section defines properties and behaviors common to all STIX Domain Objects, Relationship Objects, and the Marking Definition Object.

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Type** | **Description** |
| **type** (required) | string | The **type** property identifies the type of STIX Object (SDO, SRO, etc). The value of the **type** field **MUST** be one of the types defined by a STIX Object (e.g., indicator). |
| **id** (required) | identifier | The **id** property universally and uniquely identifies this object. All objects with the same **id** are considered different versions of the same object.  Because the object type is part of the identifier, it is not possible for objects of different types to share the same **id**. |
| **created\_by\_ref** (optional) | identifier | The ID of the Source object that describes who created this object. |
| **created** (required) | timestamp | The **created** property represents the time at which the first version of this object was created. The object creator **SHOULD** use the time it deems most appropriate as the time the object was created.  The **created** property **SHOULD** be the same across all versions of the object unless the **created** property itself was corrected by some version. |
| **modified** (required) | timestamp | The **modified** property represents the time that this particular version of the object was created. The object creator **SHOULD** use the time it deems most appropriate as the time this version of the object was created. The value of the **modified** property for a given object version **MUST** be later than or equal to the value of the **created** property.  The **modified** property **MUST** be updated when a new version of an object is issued. See section TODO for more information about versioning STIX objects. |
| **version** (required) | number | The **version** property indicates the version of this object. This field’s value **MUST** be an integer (whole number) greater than or equal to 1 and less than or equal to 999,999,999. Higher numbers indicate later versions of the object. Object creators **MUST** increase the version number (**SHOULD** increment it by exactly 1) when creating a new version of an object. See section TODO for more information about versioning STIX objects. |
| **version\_comment** (optional) | string | A comment outlining why this version of this object was created. See section TODO for more information about versioning STIX objects. |
| **revoked** (optional) | boolean | The **revoked** property indicates whether the object has been revoked. Revoked objects are no longer considered worthwhile intelligence. Revoking an object is permanent; future versions of the object with this **id** **MUST NOT** be created. See section TODO for more information about versioning STIX objects. |
| **labels** (optional) | list of type string | This field specifies a set of classifications, sometimes with a suggested open-vocab, and allows for implementation-dependent or trust group-dependent labels or tags to be applied to this object for further classification and sorting.  This field usually includes a suggested vocabulary and items in this list **SHOULD** come from that vocabulary. Additional labels **MAY** be added beyond what is in the open / suggested vocabulary based on needs and requirements of implementations and trust groups. |
| **external\_references**  (optional) | list of type external-reference | A list of external references which refers to non-STIX information. This field **MAY** be used to provide one or more URLs, descriptions, or IDs to records in other systems. |
| **object\_marking\_refs** (optional) | list of type identifier | The list of marking-definition objects to be applied to this object. See the Data Markings in section TODO for further information. |
| **granular\_markings** (optional) | list of type granular-marking | The set of granular markings applied to this object. See the Data Markings in section [TODO Ref] for further information. |

## ​8.2.​ IDs and References

The **id** property universally and uniquely identifies an object. It **MUST** conform to the identifier type.

The STIX language makes use of identifiers as defined by the identifier type for all STIX Objects. The identifier type is also used to define fields that are *ID references* to other constructs (such as the **created\_by\_ref** property in all STIX Objects). *Resolving*an ID reference is the process of identifying and obtaining the actual object referred by the ID reference field. ID references resolve to an object when the value of the ID reference property (e.g., **created\_by\_ref**) is an exact match with the **id** property of another object. ID references **MAY** refer to objects to which the consumer may not currently have.

## ​8.3.​ Object Creator

The object creator is the entity (e.g., system, organization, instance of a tool) that generates the **id** property for a given object. In STIX, object creators are represented as Source objects. A reference to the Source object representing the object creator is captured in the **created\_by\_ref** property.

Entities that re-publish an object from another entity without making any changes to the object, and thus maintaining the original **id**,are not considered the object creator and **MUST NOT** change the **created\_by\_ref** property. An entity that accepts objects and republishes them with modifications or omissions **MUST** create a new **id** for the object as they will be considered the object creator of the new object for purposes of versioning.

## ​8.4.​ Versioning

STIX content is versioned using the **version**, **version\_comment**, **revoked**, **created**, and **modified** properties. See the properties table in Section TODO [add reference] for full definitions and normative usage of the individual properties; this section describes the broader versioning process and normative rules for performing versioning and revocation.

STIX Objects can be versioned in order to update, add, or remove information. All STIX representations of objects are actually representations of one version of that object, identified in the **version** property. Higher values of the **version** property indicate later versions of the object.

STIX Objects have a single *object creator*: the entity that generates the **id** for the object and creates the first version. Only the object creator is permitted to create new versions of a STIX Object. Producers other than the object creator **MUST NOT** create new versions of that object. If a producer other than the object creator wishes to create a new version, they instead **MUST** create a new object with a new **id**. They **SHOULD** additionally create a derived-from Relationship object to relate their new object to the original object that it was derived from.

Individual versions of STIX Objects are immutable: representations of a given version of an object (identified by the object's **id** and its **version**) **MUST**, in all representations, always have the same set of properties and the same values for each property. In order to change the value of any property, or to add or remove properties, the **version** number **MUST** be increased to indicate a new version and the modified property **MUST** be updated to reflect the time of the change.

Objects can also be revoked, which is an indication that they are no longer considered actionable or worthwhile intelligence. As with issuing a new version, only the object creator is permitted to revoke a STIX Object. A value of true in the **revoked** property indicates that an object has been revoked. Revocation is permanent: once an object is marked as revoked, later versions of that object **MUST NOT** be created. The change to the **revoked** property to indicate that an object is revoked is an update to the object, and therefore its **version** and **modified** properties **MUST** be updated.

### ​8.4.1.​ Versioning Timestamps

*Non-Normative Section*

There are two timestamp properties used to indicate when STIX Objects were created and modified: **created** and **modified**. The **created** property indicates the time the first version of the object was created. The **modified** property indicates the time the current version of the object (indicated by the value of the **version** property) was created. For both the **created** and **modified** properties, object creators may use any time that best represents when they want to say the object was created and modified. Some creators might use the time the representation was created in their own database; other creators might use the time the object was first shared externally; and others might use a user-tunable value.

### ​8.4.2.​ New Version or New Object?

*Non-Normative Section*

Eventually an implementation will encounter a case where a decision must be made regarding whether a change is a new version of an existing object or is different enough that it is a new object. This is generally considered a data quality problem and therefore this specification does not provide any normative text.

However, to assist implementers and promote consistency across implementations, some rules of thumb are provided. Any time a change indicates a *material change* to the meaning of the object , a new object with a different **id** should be used. A material change is any change that the object creator believes substantively changes the meaning of the object. As an example, an object creator might consider changing a Threat Actor from one country to another a material change. These decisions are always made by the object creator. The creator should also think about references to the object when deciding if a change is material. If the change would invalidate the usefulness of references to the object, then the change is material and a new object **id** should be used.

### ​8.4.3.​ Examples

**Example Version**

One object creator has decided that the previous title they used for a SDO is incorrect. They consider that change as an update to the object.

|  |  |  |
| --- | --- | --- |
| **Step #** | **STIX Object** | **Object Creator Action** |
| 1 | {  "type": "example",  "id": "example--1",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-01T06:13:14.000000Z",  "version": 1,  "title": "attention",  "description": "this is the description"  } | Original version of an object is created. |
| 2 | ...  "title": "Attention!",  ... | Object creator changes the title. |
| 3 | {  "type": "example",  "id": "example--1",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-08T03:43:44.000000Z",  "version": 2,  "title": "Attention!",  "description": "this is the description"  } | Object creator increases current object **version** by 1 and updates the **modified**. |

**Example of Derived Object**

One object creator has decided that the previous title they used for a SDO is incorrect. They consider that change fundamental to the meaning of the object and therefore revoke the object and issue a new one.

1. A revised initial indicator, with the same **id**, updated **version**, and the **revoked** flag set.
2. A new indicator with a new **id.**

|  |  |  |
| --- | --- | --- |
| **Step #** | **STIX Object** | **Object Creator Action** |
| 1 | {  "type": "example",  "id": "example--1",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-01T06:13:14.000000Z",  "version": 1,  "title": "attention",  "description": "this is the description"  } | Original object created (via new id and set **version** to *1*). |
| 2 | N/A, STIX is not involved in this step | Object creator changes the title in their internal database. |
| 3 | {  "type": "example",  "id": "example--1",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-08T03:43:44.000000Z",  "version": 2,  "title": "attention",  "description": "this is the description",  "revoked": true  } | Object creator revokes the existing object by setting **revoked** to true. |
| 4 | {  "type": "example",  "id": "example--2",  "created": "2016-05-08T03:43:44.000000Z",  "modified": "2016-05-08T03:43:44.000000Z",  "version": 1,  "title": "Attention!",  "description": "this is the description"  } | Object creator creates a new object (with a new **id** and **version** set to *1*). |
| 5 | {  "type": "relationship",  "id": "relationship--3",  "created": "2016-05-08T03:43:44.000000Z",  "modified": "2016-05-08T03:43:44.000000Z",  "version": 1,  "source\_ref": "example--1",  "target\_ref": "example--2",  "name": "derived-from"  } | (Optional) Object creator creates a new Relationship indicating that the new object is derived from the old object. |

**Example Recipient Workflow**

This section describes an example workflow where a recipient receives multiple updates to a particular object. (In this example, the STIX Objects have been truncated for brevity.)

|  |  |  |
| --- | --- | --- |
| **Step #** | **STIX Object** | **Recipient Action** |
| 1 | {  "type": "example",  "id": "example--1",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-01T06:13:14.000000Z",  "version": 1  } | Recipient stores example object because this is the first time the recipient has seen the object. |
| 2 | {  "type": "example",  "id": "example--1",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-08T03:43:44.000000Z",  "version": 4  } | Recipient updates example object because the received version number is higher than the object that is currently stored. |
| 3 | {  "type": "example",  "id": "example--1",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-06T06:23:45.000000Z",  "version": 3  } | Recipient ignores this object because the recipient already has a newer version of the object.  Note: recipient might choose to store meta-information about received objects, including versions that were received out-of-order. |
| 4 | {  "type": "example",  "id": "example--1",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-11T06:41:21.000000Z",  "version": 12,  "revoked": true  } | Recipient deletes example object, but keeps some metadata regarding the object. |
| 5 | {  "type": "example",  "id": "example--1",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-10T17:28:54.000000Z",  "version": 11  } | Recipient ignores this object because the recipient already has a newer version of the object (the revoked version). |

​

**Example Object Creator Workflow**

This section describes an example workflow where a object creator publishes multiple updates to a particular object. This scenario assumes a human using a STIX implementation. (In this example, the STIX Objects have been truncated for brevity.)

|  |  |  |
| --- | --- | --- |
| **Step #** | **STIX Object** | **User Action** |
| 1 | N/A – STIX is not involved in this scenario.  (Tools *could* choose to create and track STIX versions for internal changes, but it is not required by the specification.) | User clicks a create button in the user interface, creates a SDO, then clicks save. This action causes information to be stored in the product’s database. |
| 2 | {  "type": "example",  "id": "example--2",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-01T06:13:14.000000Z",  "version": 1  } | The user clicks the “share” button, delivering the intelligence to sharing partners. |
| 3 | N/A – STIX is not involved in this scenario.  (Tools *could* choose to create and track STIX versions for internal changes, but it is not required by the specification.) | The user performs additional analysis within the STIX implementation, performing multiple modifications and saving their work multiple times. |
| 4 | {  "type": "example",  "id": "example--2",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-03T16:33:51.000000Z",  "version": 2  } | The user, happy with the status of their work, decides to provide an update to some properties of the previously published object (not shown). |
| 5 | {  "type": "example",  "id": "example--2",  "created": "2016-05-01T06:13:14.000000Z",  "modified": "2016-05-08T13:35:12.000000Z"  "version": 3,  "revoked": true,  } | The user receives lots of negative feedback regarding the quality of their work and decides to retract the object by pressing the “revoke” button. |

## ​8.5.​ Common Relationships

Each SDO has its own set of relationships that are specified in the definition of that SDO. Common Relationships are relationships that are defined for all STIX Objects. See Section <to do>[add reference] for more information about relationships.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Source** | **Target** | **Description** |
| derived-from | *<STIX Object>* | *<STIX Object of same type>* | STIX objects can be derived from other objects, which means that the object contains information from the related object. Derivation is explicitly a relationship between two separate objects and **MUST NOT** be used as a substitute for the normal object versioning process. |
| duplicate-of | *<STIX Object>* | *<STIX Object of same type>* | Allows recording that two different Objects of the same type are duplicates of each other and one should be disregarded or deleted. There is no suggestion of which one is preferred implicit in this relationship source and target.  As an example, a Campaign object from one organization could be marked as a duplicate-of a Campaign object from another organization if they both described the same actual campaign. |
| related-to | *<STIX Object>* | *<STIX Object of any type>* | The catch-all generic relationship type that allows description of relationships between an Object and any other Domain Object. It is essentially a fallback to allow a relationship to be defined when none of the others fit.  As an example, a Malware object describing a piece of malware could be marked as a related-to a Tool if they are commonly used together. That relationship is not common enough to standardize on, but may be useful to some analysts. |

## ​8.6.​ Reserved Properties

This section defines property names that are reserved for future use in revisions of this document. The property names defined in this section **MUST NOT** be used for the name of any Custom Property.

Properties that are currently reserved across all STIX Objects are:

* confidence
* severity

Specific properties per SDO that are currently reserved are:

* Course of Action SDO: action
* Source, Threat Actor, Victim SDOs: usernames, phone\_numbers, addresses

# ​9.​ Data Handling in STIX

Data markings represent restrictions and permissions or other guidance for how that data can be used and shared. For example, data may be shared with the restriction that it must not be re-shared, or that it must be encrypted at rest. In STIX, data markings are specified using the marking-definition object. Object markings and granular markings apply marking-definition objects to other STIX Objects and properties, respectively.

Some types of marking definitions or trust groups have rules about which markings have precedence over other markings or which markings can be additive to other markings. However, the STIX specification does not define precedence or which marking definitions should have precedence over other definitions.

## ​9.1.​ Marking Definition

**Type Name:** marking-definition

The marking-definition object represents a specific instance of a marking, defined by an external marking system (e.g., the traffic light protocol, TLP) and contained in the definition property. Data markings typically capture handling or sharing requirements for data, and are applied in the **object\_markings\_refs** and **granular\_markings** properties on STIX Objects, which reference a list of IDs for marking-definition objects. The **object\_markings\_refs** and **granular\_markings** properties of a marking-definition object **MUST NOT** contain references to itself, i.e. cannot contain circular references.

### ​9.1.1.​ Properties

|  |  |  |
| --- | --- | --- |
| **Common Properties** | | |
| **type, id, created\_by\_ref, created, modified, version, revoked, version\_comment, labels, external\_references, object\_markings\_refs, granular\_markings** | | |
| **Property Name** | **Type** | **Description** |
| **type** (required) | string | The value of this field **MUST** be marking-definition. |
| **definition** (required) | <marking object> | The detailed information of the marking. This field **SHOULD** be of type: tlp-marking or statement-marking. Other marking definitions **MAY** be used. |

### ​9.1.2.​ Relationships

These are no relationships explicitly defined between the Marking Definition object and other objects, other than those defined as common relationships. The first section lists the embedded relationships by property name along with their corresponding target.

Relationships are not restricted to those listed below. Relationships can be created between any objects using the related-to relationship name or, as with open vocabularies, user-defined names.

|  |  |  |  |
| --- | --- | --- | --- |
| **Embedded Relationships** | | | |
| **created\_by\_ref** | | source | |
| **object\_markings\_refs** | | marking-definition | |
| **Common Relationships** | | | |
| duplicate-of, derived-from, related-to | | | |

​

### ​9.1.3.​ TLP Marking Object Type

The TLP marking type sub-object defines how you would represent a TLP marking in a definition field.

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Type** | **Description** |
| **type** (required) | string | The value of this field **MUST** be tlp-marking. |
| **tlp** (required) | tlp-marking-enum | The TLP level (defined by FIRST, ask Tom Millar for stable ref) of the content marked by this marking definition. |

The following standard definitions **MUST** be used to reference or represent TLP markings. Other instances of tlp-marking **MUST NOT** be used.

|  |  |
| --- | --- |
| white | {  "type": "marking-definition",  "id": "marking-definition--613f2e26-407d-48c7-9eca-b8e91df99dc9",  "created": "2016-08-01T00:00:00Z",  "modified": "2016-08-01T00:00:00Z",  "version": 1,  "definition": {  "type": "tlp-marking",  "tlp": "white"  }  } |
| green | {  "type": "marking-definition",  "id": "marking-definition--34098fce-860f-48ae-8e50-ebd3cc5e41da",  "created": "2016-08-01T00:00:00Z",  "modified": "2016-08-01T00:00:00Z",  "version": 1,  "definition": {  "type": "tlp-marking",  "tlp": "green"  }  } |
| amber | {  "type": "marking-definition",  "id": "marking-definition--f88d31f6-486f-44da-b317-01333bde0b82",  "created": "2016-08-01T00:00:00Z",  "modified": "2016-08-01T00:00:00Z",  "version": 1,  "definition": {  "type": "tlp-marking",  "tlp": "amber"  }  } |
| red | {  "type": "marking-definition",  "id": "marking-definition--5e57c739-391a-4eb3-b6be-7d15ca92d5ed",  "created": "2016-08-01T00:00:00Z",  "modified": "2016-08-01T00:00:00Z",  "version": 1,  "definition": {  "type": "tlp-marking",  "tlp": "red"  }  } |

### ​9.1.4.​ Statement Marking Object Type

The statement marking type sub-object defines how you would represent a text marking statement (copyright, terms of use, etc.) in a definition. Statement marking types do not override each other.

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Type** | **Description** |
| **type** (required) | string | The value of this field **MUST** be statement-marking. |
| **statement** (required) | string | A statement (e.g., copyright, terms of use) applied to the content marked by this marking definition. |

​

### ​9.1.5.​ Examples

{

"type": "marking-definition",

"id": "marking-definition--34098fce-860f-48ae-8e50-ebd3cc5e41da",

"created": "2016-08-01T00:00:00Z",

"modified": "2016-08-01T00:00:00Z",

"version": 1,

"definition": {

"type": "tlp-marking",

"tlp": "green"

}

}

{

"type": "marking-definition",

"id": "marking-definition--089a6ecb-cc15-43cc-9494-767639779124",

"created": "2016-08-01T00:00:00Z",

"modified": "2016-08-01T00:00:00Z",

"version": 1,

"definition": {

"type": "statement-marking",

"statement": "Copyright 2016, Example Corp"

}

}

{

"type": "indicator",

"id": "indicator--089a6ecb-cc15-43cc-9494-767639779235",

...

"object\_marking\_refs": [

"marking-definition--089a6ecb-cc15-43cc-9494-767639779124",

"marking-definition--34098fce-860f-48ae-8e50-ebd3cc5e41da"

],

...

}

## ​9.2.​ Object Markings

Object Markings apply data markings to an entire STIX Object and all of its contents. Object Markings are specified in the **object\_marking\_refs** property, which is an optional list of ID references (of type identifier) that resolve to objects of type marking-definition. The markings referenced in the **object\_marking\_refs** property and defined in the marking-definition object apply to that STIX Object and all of its contents. Changes to the **object\_marking\_refs** property (and therefore the markings applied to the object) are treated the same as changes to any other properties on the object and follow the same rules for versioning.

### ​9.2.1.​ Examples

This example marks the indicator and all its properties with the marking definition referenced by the ID.

{

"type": "indicator",

"id": "indicator--089a6ecb-cc15-43cc-9494-767639779235",

...

"object\_marking\_refs": ["marking-definition--089a6ecb-cc15-43cc-9494-767639779123"],

...

}

## ​9.3.​ Granular Markings

Whereas object markings apply to an entire STIX Object and all its properties, granular markings allow data markings to be applied to individual portions of STIX Objects. Granular markings are specified in the **granular\_markings** property, which is a list of granular-marking instances. Each of those instances contains a list of selectors to indicate what is marked and a list of references to the marking-definition objects to be applied. Granular markings can be used, for example, to indicate that the **name** property of an indicator should be handled as TLP:GREEN, the **description** property as TLP:AMBER, and the **pattern** property as TLP:RED.

### ​9.3.1.​ Granular Marking Type

The granular-marking type defines how the list of marking-definition objects referenced by the **marking\_refs** property to apply to a set of content identified by the list of selectors in the **selectors** property.

|  |  |  |
| --- | --- | --- |
| **Property Name** | **Type** | **Description** |
| **marking\_refs** (required) | list of type identifier | The list of markings that apply to the fields selected by **selectors**. |
| **selectors**  (required) | list of type string | A list of selectors for content contained within the STIX Object in which this field appears. Selectors **MUST** conform to the syntax defined in [Section TODO].  The markings referenced in the **marking\_refs** field are applied to the content selected by the selectors in this list. |

#### ​9.3.1.1.​ Selector Syntax

Selectors contained in the **selectors** list are strings that consist of multiple components that **MUST** be separated by the . character. Each component **MUST** be one of:

* A property name, e.g. description, or;
* A zero-based array index, specified as a non-negative integer in square brackets, e.g. [4]

Selectors are path traversals: the root of each selector is the STIX Object that the **granular\_markings** field appears in. Starting from that root, for each component in the selector, properties and array items are traversed. When the complete list has been traversed, the value of the content is considered selected.

Selectors **MUST** refer to properties or array items that are actually present on the marked object.

As an example, consider the following STIX Object:

{

"type": "example",

"description": "hello",

"extra": {

"foo": "some field",

"bar": "some other field"

},

"widgets": ["first", "second"]

}

Valid selectors:

* description selects the **description** property ("hello")
* extra.foo selects the **foo** property contained within the **extra** property ("some field")
* widgets.[0] selects the first item contained within the **widgets** list ("first")
* widgets selects the list contained in the **widgets** property. Due to the recursive nature of the selector, that includes all items in the list (["first", "second"]).
* extra selects the object contained in the **extra** property. Due to the recursive nature of the selector, that includes both the **foo** and **bar** properties.

Invalid selectors:

* foobar and widgets.[3] are invalid selectors because they refer to content not present in that object.
* extra.[0] is an invalid selector because the extra property is an object and not a list.
* widgets.name is an invalid selector because widgets property is a list and not an object.

*Non-Normative Text*

This syntax is inspired by JSONPath and is in fact a strict subset of allowable JSONPath expressions (with the exception that the '$' to indicate the root is implicit). Care should be taken when passing selectors to JSONPath evaluators to ensure that the root is correct. It is expected, however, that selectors can be easily evaluated in programming languages that implement list and key/value mapping types (dictionaries, hashmaps, etc.) without resorting to an external library.

### ​9.3.2.​ Example

This example marks the **description** and **labels** properties with the single marking definition referenced in the list.

{

...

"granular\_markings": [

{

"selectors": ["description", "entries"],

"marking\_refs": ["marking-definition--089a6ecb-cc15-43cc-9494-767639779123"]

}

],

"description": "Some description"

"title": "Some title",

"labels": ["first", "second"]

}

# ​10.​ STIX Domain Objects

The STIX Domain Objects will go here when we merge the documents.

# ​11.​ Relationship Objects

The Relationship Objects will go here when we merge the documents.

# ​12.​ Metadata Objects

The Metadata Objects will go here when we merge the documents.

# ​13.​ Vocabularies

STIX supports two types of vocabularies:

* **Open vocabularies** (indicated throughout this specification as ‘ov’), which provide a listing of common and industry accepted terms as a guide to the user but do not limit the user to that defined list; and
* **Enumerations**, (indicated throughout this specification as 'enum'), which provide a hardcoded list of terms defined by the specification that **MUST NOT** be expanded by implementations. Future versions of STIX may expand or change these enumerations.

The following sections provide object-specific listings for each of the vocabularies referenced in the object description sections.

## ​13.1.​ Attack Motivation

**Type Name:** attack-motivation-ov

This vocabulary is currently used in the following SDOs in the STIX model:

* Campaign
* Intrusion Set
* Threat Actor

Knowing a Threat Actor's or Campaign's motivation narrows which targets that actor or campaign may focus on. For example, crime syndicates, who have a strong profit motive, will generally only take assets they can easily convert to cash regardless of the discoverability of their actions, while Threat Actors seeking notoriety will ignore attacks on non-visible assets that will not bring them attention.

Understanding the Threat Actors’ or Campaign's intent helps defenders focus their often-limited defense resources on the most likely attack scenarios for any particular asset.

Motivation shapes the intensity and the persistence of an attack. Threat Actors and Campaigns usually act in a manner that reflects their underlying emotion or situation, and this informs defenders of the manner of attack. For example, a spy motivated by nationalism (ideology) likely has the patience to achieve long-term goals and work quietly for years, whereas a cyber-vandal out for kicks can create an intense and attention-grabbing attack but quickly loses interest and moves on. Understanding these differences allows defenders to implement controls tailored to each type of attack for greatest efficiency.

Motivation helps in describing threat and risk scenarios in less technical terms. Analysts must eventually convey all Threat Actor analysis to others in their organization who can act to help mitigate the risks. Describing Attack Motivation tells a fuller, more relatable story to colleagues of all security levels. Communicating risks in a more understandable fashion obviously leads to faster implementation of more effective defenses.

This section including vocabulary items and their descriptions is based on and contains copied text from the Threat Agent Motivations publication from Intel Corp in Feb 2015[[1]](#footnote-0)

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| accidental, coercion, dominance, ideology, notoriety, organizational-gain, personal-gain, personal-satisfaction, revenge, unpredictable | |
| **Vocabulary Value** | **Description** |
| accidental | Benevolent or harmless intent but with actions that inadvertently cause harm.  This element generally describes the non-hostile Threat Actor, such as a well-meaning, dedicated employee who through distraction or poor training unintentionally causes harm to his or her organization. A common occurrence is an employee who was quickly assigned additional duties to cover for laid-off employees but has not yet received proper training. With a heavy workload and lacking a full understanding of the tasks, the employee is bound to make mistakes, unwittingly and possibly without even knowing a mistake has occurred. |
| coercion | Forced to act illegally on behalf of another.  Unlike the other Motivations, a coerced person does not act for personal gain, but out of fear of incurring a loss. These individuals have been forced through intimidation or blackmail to act for someone else’s benefit and are conducting acts they probably would not normally do and that may even directly conflict with their own self-interests. In most cases, a coerced person is just as much a victim as the attack target.  Coercion can effectively force a person  to commit very harmful, possibly violent actions, if the threat against him or her is severe enough. Coercion can also subvert employees often considered above reproach, such as executives or those who undergo regular security checks.  There are probably fewer total Threat Actors driven by coercion than by the other motivations, but it can be a motivator for almost any kind of threat and must be considered when analyzing and planning for risks. |
| dominance | Attempting to assert superiority over another.  Dominance can take many forms at many scales, for example, physically intimidating a coworker, threatening to expose sensitive data of a corporation, or amassing an army along a border. But in all cases Threat Actors use whatever power they have to bully others into submission.  Threat Actors seeking dominance may also steal information assets to create power and build toward a goal of dominance. Collection can include compromising items such as sensitive intellectual property, personal information, business data, product data, and information on operational aspects such as networks and supply chains. Access to these items allows an attacker to leverage them or their vulnerabilities during an attack. For example, to prepare for a cyber attack during a future national conflict, a government spy may steal software bug reports from a network device manufacturer, which detail the device’s vulnerabilities and enable cyberattacks.  Ideology and Dominance may both be present in some state-sponsored Threat Actors, but Dominance can occur with or without Ideology. Crime Syndicates often act to establish dominance in extreme acts of bullying but not in support or in conjunction with some higher objective—that is, Ideology.  Vandalism and hacking are also included under Dominance, because cyber vandals typically seek dominance over others through bullying. Other factors, such as Notoriety, may also be present to some degree in these Threat Actors, but Dominance is the primary motivator. |
| ideology | A passion to express a set of ideas, beliefs, and values that shapes and drives harmful acts.  Threat Actors who act for ideological reasons (e.g. political, religious, human rights, environmental, etc.) are not usually motivated primarily by the desire for profit; they are acting on their own sense of morality, justice, or political loyalty. Ideological motivation can arise independent of any prior interaction with the target. For example, an activist group may sabotage a company’s equipment because they believe the company is harming the environment even though the activists may have never actually used any of the company’s products.  Because ideologies vary, so do the types of threat posed by individuals or organizations with this motivation. The threat may come in the form of a direct attack, such as sabotage, theft, or exposure of sensitive information. It may also happen indirectly, such as an employee who improperly uses company computers to participate in a cyberattack against an organization the employee believes to be oppressive. If traced back, the attacked organization could launch a counterattack or bring legal action against the unsuspecting “attacking” company. |
| notoriety | Seeking to become well known for harmful activity.  Threat Actors motivated by Notoriety are often seeking either personal validation or respect within a community. The actions used to achieve even unreasonable notoriety may be quite well reasoned and strategic. Similar to vandalism, the individual or group may seek to cause damage for its own sake, but staying covert is not a priority—quite the opposite, in fact. To garner the respect of their target audience, the actions that those seeking notoriety take are not tempered by a need for secrecy and therefore can be extreme in scope and damage. |
| organizational-gain | Seeking an advantage over a competitor’s organization.  The prospect of increased profit or other gains through an unfairly obtained competitive advantage has always been a powerful incentive, and the temptation to cheat will always be too strong for some to resist. Through theft of information such as intellectual property, business processes, or supply chain agreements, a competitor bypasses the lengthy and expensive process of developing it themselves, accelerating its position in a market or capability. The inappropriate acquisition or misuse of information, even seemingly esoteric data such as employee demographics, could be used to gain a competitive edge.  Information theft is not the only option used to get ahead. A competitor could also choose sabotage, lawsuits, or other non-theft means to undermine a competing organization to gain an advantage.  Organizational Gain includes military objectives as well. A military organization can use stolen information to advance its own technology while also enabling careful study of their target’s capabilities and vulnerabilities.  In some cases, individuals with similar objectives may work collaboratively to advance their own personal gain but do so under voluntary organizational rules, such as military mercenaries or hacktivist collectives. In these cases, the organization motivation reflects the individual's’ motivation. |
| personal-gain | Improve one’s own financial status.  A selfish desire for personal gain motivates many crimes. This element describes individuals who steal money in some way or conduct activities that will net them money, such as hacking in exchange for a paycheck. These individuals are most likely indifferent to the damage caused by their actions, but apart from stealing, usually do not go out of their way to harm their target.  This motivation is different from Organizational Gain in the timeliness of the threat. Usually, the individual stealing assets wants to make a quick profit by selling them, rather than invest the time and expertise needed to craft a package for sale like an organization might create. Financial fraud is a result of this motivation, as is physical theft of valuable items. Intellectual property theft for sale is also a result of this motivation, but in the special case of espionage, Ideology may also play a significant part in an individual’s motivation.  An individual Threat Actor may be seeking personal gain, but this does not mean that the Threat Actor always acts alone. Many criminal groups, organized or not, are often made up of individuals banded together solely to maximize their own personal profits.  In addition to greed, the need to steal can stem from other factors, such as pressing medical or addiction debts, poverty, coercion, disgruntlement, or mental impairment. These issues can easily lead an otherwise honest individual to commit illegal acts.  Personal Financial Gain can also apply to individuals working for an organizational Threat Actor, such as a Competitor or Organized Crime. While the organization seeks an advantage for its collective goals, the individuals working for that organization may be driven by more personal reasons that may have little to do the organization’s objectives. Often this personal motivation is simply the personal financial gain that results from supporting the organization, such as a paycheck or a cut of the spoils. (See the Personal Motivation modifier). |
| personal-satisfaction | Fulfilling an emotional self-interest.  Some people may cause harm when they act not to support a financial or ideological objective but to satisfy a strictly internal, personal interest. This personal interest can be expressed in many ways, such as intrusive curiosity or thrill seeking, like children who break into a building just for the excitement of going where they are not allowed. More harmful possibilities include a healthcare worker who inappropriately reads the medical records of celebrities to see what treatment they are receiving or a hacker who attacks a website primarily because he or she enjoys the lawlessness of the act. Most “crimes of passion”—those caused by love, anger, fear, and so on— also fall under Personal Satisfaction.  Threat Actors driven by Personal Satisfaction may incidentally receive some other gain from their actions, such as a profit, but their primary motivation is to gratify a personal, emotional need. This personal interest does not preclude people from banding together with other like-minded individuals toward a mutual, but not necessarily organizational, objective. |
| revenge | A desire to avenge perceived wrongs through harm.  Most people go through stages of dissatisfaction with their employer or with a company they have done business with, but usually the situation resolves without illegal behavior. When the grievance (real or perceived) is severe and the situation escalates, a disgruntled person can seek revengeful and harmful retaliation. Unlike Ideology, Disgruntlement implies there is a history of some direct interaction with the target organization.  A disgruntled Threat Actor seeking Revenge can include employees or former employees, all of whom may have extensive knowledge that the actors can leverage when conducting attacks. Often a disgruntled individual acts alone but may join an organization, whether a competitor, group of similar individuals, or criminal organization, if the individual believes that doing so will enable him or her to better harm the source of his or her anger.  Predicting the actions of a disgruntled person or group is very difficult, because the action can take many forms including sabotage, violence, theft, fraud, or embarrassing individuals or the organization. |
| unpredictable | Acting without identifiable reason or purpose and creating unpredictable events.  It may seem that since Unpredictable represents the actions one cannot anticipate, there is no need to include it—everyone knows life has many surprises. However, explicitly recognizing the potential for unpredictability in an environment and comprehending it in planning is essential for effective risk management. We include Unpredictable here to enable and support the discipline of managing the unexpected.  In its application as a Threat Actor Motivation, Unpredictable is not a miscellaneous or default category. It does not include acts such as a sudden DDOS attack on a company website or a new type of email phishing campaign. Those events may have occurred unexpectedly and the methods may have been novel, but a reasonable person could easily anticipate those kinds of events would occur at some point. In this taxonomy, Unpredictable means a truly random and likely bizarre event, which seems to have no logical purpose to the victims. |

## ​13.2.​ Attack Objective

**Type Name:** attack-objective-ov

This vocabulary is currently used in the following SDOs in the STIX model:

* Campaign
* Intrusion Set
* Threat Actor

Attack Objective is an open vocabulary that captures the objectives of a particular attacker or series of attacks (campaign or intrusion set). The vocabulary is divided into two sections: the high-level objectives come from the Intel Threat Agent Library publication and describe the broad objectives that the attacker wants to achieve. The specific means capture the mechanisms by which those objectives are achieved. For example, an attacker interested in damage might choose to cause damage to an organization by damaging their brand.

This vocabulary captures both types of objectives to simplify usage and avoid divergence in practice.

This section including vocabulary items and their descriptions is based on and contains copied text from the Threat Agent Library publication from Intel Corp in Sept 2007[[2]](#footnote-1)

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| acquisition-theft, business-advantage, damage, embarrassment, technical-advantage | |
| **Vocabulary Value** | **Description** |
| **High-Level Objectives** | *The broad objectives that the attacker wishes to achieve.* |
| acquisition-theft | Illicit acquisition of valuable assets, for example for resale or extortion, in a way that preserves the assets’ integrity but may incidentally damage other items in the process. |
| business-advantage | Increased ability to compete in a market with a given set of products. The goal is often to acquire business information, processes, or assets such as inventory control processes, acquisition plans, customer lists, marketing research, or legal proceedings. |
| damage | Deliberate injury to personnel, physical or electronic assets, or intellectual property. The goal is often to cause terror, interfere with or stop the organization's operations, or incur significant costs to the organization. |
| embarrassment | Public portrayal in an unflattering light, causing a loss of influence, credibility, competitiveness, or brand or stock value. This will commonly result from exposure of sensitive information, but may also involve disproving an organization's product claims or values. |
| technical-advantage | Illicit improvement of a specific product or production capability. The primary target is to acquire production processes or assets rather than a business process. Common targets are manufacturing processes, intellectual property, design personnel, customer specifications, and industrial control systems operations. |
| **Specific Means / Goals** | *The specific means by which the attacker attempts to achieve the above high-level objectives.* |
| account-takeover | Obtain control over an account (financial, etc) |
| credential-theft | Theft of credentials in bulk |
| degradation-of-service | Reducing the level of services provided by the target |
| extortion | Force the payment of some sort to prevent the attacker from taking some action. |
| harassment | Pressure or intimidate the target |
| identity-theft | Theft of the target’s identity |
| intellectual-property-theft | Theft of intellectual property |

## ​13.3.​ Attack Resource Level

**Type Name:** attack-resource-level-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Campaign
* Intrusion Set
* Threat Actor

Attack Resource Level is an open vocabulary that captures the general level of resources that a threat actor, intrusion set, or campaign might have access to. It ranges from individual, a person acting alone, to government, the resources of a national government.

This section including vocabulary items and their descriptions is based on and contains copied text from the Threat Agent Library publication from Intel Corp in Sept 2007[[3]](#footnote-2)

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| individual, club, contest, team, organization, government | |
| **Vocabulary Value** | **Description** |
| individual | Resources limited to the average individual; Threat Actor acts independently. Minimum Sophistication level: None. |
| club | Members interact on a social and volunteer basis, often with little personal interest in the specific target. An example might be a core group of unrelated activists who regularly exchange tips on a particular blog. Group persists long term. Minimum Sophistication level: Novice. |
| contest | A short-lived and perhaps anonymous interaction that concludes when the participants have achieved a single goal. For example, people who break into systems just for thrills or prestige may hold a contest to see who can break into a specific target first. It also includes announced "operations" to achieve a specific goal, such as the original "OpIsrael" call for volunteers to disrupt all Israel internet functions for a day. Minimum Sophistication level: Practitioner. |
| team | A formally organized group with a leader, typically motivated by a specific goal and organized around that goal. Group persists long term and typically operates within a single geography. Minimum Sophistication level: Practitioner. |
| organization | Larger and better resourced than a Team; typically a company or crime syndicate. Usually operates in multiple geographies and persists long term. Minimum Sophistication level: Expert. |
| government | Controls public assets and functions within a jurisdiction; very well resourced and persists long term. Minimum Sophistication level: Expert. |

## ​13.4.​ Attack Sophistication Level

**Type Name:** attack-sophistication-level-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Threat Actor

The attack sophistication vocabulary captures the skill level of a threat actor. It ranges from "none", which describes a complete novice, to "innovator", which describes an actor who is able to create their own types of attacks and discover 0-days. This vocabulary is separate from resource level: an innovative, highly-skilled threat actor may have access to very few resources while a practitioner-level actor might have the resources of an organized crime ring.

This section including vocabulary items and their descriptions is based on and contains copied text from the Threat Agent Library publication from Intel Corp in Sept 2007[[4]](#footnote-3)

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| none, novice, practitioner, expert, innovator | |
| **Vocabulary Value** | **Description** |
| none | Has average intelligence and ability and can easily carry out random acts of disruption or destruction, but has no expertise or training in the specific methods necessary for a targeted attack. |
| novice | Can copy and use existing techniques. Example: Untrained Employee.  Demonstrates a nascent capability. A novice has basic computer skills and likely requires the assistance of a Practitioner or higher to engage in hacking activity. They use existing and frequently well known and easy-to-find techniques and programs or scripts to search for and exploit weaknesses in other computers on the Internet and lack the ability to conduct their own reconnaissance and targeting research. |
| practitioner | Has a demonstrated, albeit low, capability. A practitioner possesses low sophistication capability. They do not have the ability to identify or exploit known vulnerabilities without the use of automated tools. They are proficient in the basic uses of publicly available hacking tools, but are unable to write or alter such programs on their own. |
| expert | Expert in technology and attack methods, and can both apply existing attacks and create new ones to greatest advantage. Example: Legal Adversary.  Demonstrates advanced capability. An actor possessing expert capability has the ability to modify existing programs or codes but does not have the capability to script sophisticated programs from scratch. The expert has a working knowledge of networks, operating systems, and possibly even defensive techniques and will typically exhibit some operational security. |
| innovator | Demonstrates sophisticated capability. An innovator has the ability to create and script unique programs and codes targeting virtually any form of technology. At this level, this actor has a deep knowledge of networks, operating systems, programming languages, firmware, and infrastructure topologies and will demonstrate operational security when conducting his activities. Innovators are largely responsible for the discovery of 0-day vulnerabilities and the development of new attack techniques. |

## ​13.5.​ Course of Action Label

**Type Name:** course-of-action-label-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Course of Action

Course of Action Label is an open vocabulary used to label Courses of Action. The labels describe the general type of action that is being represented, such as redirection (for example to a honeynet), internal blocking (for example at the host level), and external blocking (for example at an external firewall). It also includes higher-level courses of action such as policy changes, diplomatic actions, and user training.

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| perimeter-blocking, internal-blocking, redirection, hardening, patching, eradication, rebuilding, training, monitoring, physical-access-restrictions, logical-access-restrictions, public-disclosure, diplomatic-actions, policy-actions | |
| **Vocabulary Value** | **Description** |
| perimeter-blocking | Perimeter-based blocking of traffic from a compromised source. |
| internal-blocking | Host-based blocking of traffic from an internal compromised source. |
| redirection | Re-routing of suspicious or known malicious traffic away from the intended target to an area where the threat can be more safely observed and analyzed. |
| hardening | Securing a system by reducing its surface of unnecessary software, usernames or logins, and running services. |
| patching | A specific form of hardening, patching involves applying a code fix directly to the software with the vulnerability. |
| eradication | Identifying, locating, and eliminating malware from the network. |
| rebuilding | Re-installing a computing resource from a known safe source in order to ensure that the malware is no longer present on the previously compromised resource. |
| training | Training users and administrators on how to identify and mitigate this type of threat. |
| monitoring | Setting up network or host-based sensors to detect the presence of this threat. |
| physical-access-restrictions | Activities associated with restricting physical access to computing resources. |
| logical-access-restrictions | Activities associated with restricting logical access to computing resources. |
| public-disclosure | Informing the public of the existence and characteristics of the threat or threat actor to influence positive change in adversary behavior. |
| diplomatic-actions | Engaging in communications and relationship building with threat actors to influence positive changes in behavior. |
| policy-actions | Modifications to policy that reduce the attack surface or infection vectors of malware. |

## ​13.6.​ Identity Classification

**Type Name:** identity-classification-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Threat Actor
* Source
* Victim Target

This vocabulary describes the classification of an identity: whether it describes an organization, group, individual, or class.

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| individual, group, organization, class, unknown | |
| **Vocabulary Value** | **Description** |
| individual | A single person. |
| group | An informal collection of people, without formal governance, such as a distributed hacker group. |
| organization | A formal organization of people, with governance, such as a company or country. |
| class | A class of entities, such as all hospitals or all Europeans. |
| unknown | It is unknown whether the classification is individual, group, organization, or class. |

## ​13.7.​ Incident Label

**Type Name:** incident-label-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Incident

Incident labels is a controlled vocabulary to categorize incidents. Items are not mutually exclusive: an incident can be both a compromise of an asset and a compromise of information.

The source for many of these vocabulary items is [TODO US-CERT REF].

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| benign, denial-of-service, improper-usage, compromise-asset, compromise-information, insider-breach, malicious-code, probing-scanning, unauthorized-access, investigating | |
| **Vocabulary Value** | **Description** |
| benign | An incident that is that is the result of an exercise, testing or a false alarm |
| denial-of-service | An attack that successfully prevents or impairs the normal authorized functionality of networks, systems or applications by exhausting resources. This activity includes being the victim or participating in the DoS |
| improper-usage | A person violates acceptable computing use policies. |
| compromise-asset | An incident that results in the compromise of an asset, such as a host, network device, application or account. |
| compromise-information | An incident that results in the disclosure, corruption or destruction of sensitive information or intellectual property. |
| insider-breach | An incident caused by a threat actor associated with the organization which was the target of the incident. |
| malicious-code | Installation of malicious software (e.g., virus, worm, Trojan horse, or other code-based malicious entity) that infects an operating system or application. |
| probing-scanning | An incident that includes any activity that seeks to access or identify a computer, open ports, protocols, service, or any combination for later exploit. This activity does not directly result in a compromise or denial of service |
| unauthorized-access | An incident in which a Threat Actor gains logical or physical access without permission to a network, system, application, data, or other resource. |
| investigating | Unconfirmed incidents that are potentially malicious or anomalous activity deemed by the reporting entity to warrant further review. |

## ​13.8.​ Indicator Label

**Type Name:** indicator-label-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Indicator

Indicator labels is an open vocabulary used to categorize Indicators. It is intended to be high-level to promote consistent practices. Indicator labels should not be used to capture information that can be better captured via related Malware or Attack Pattern objects. It is better to link an Indicator to a Malware object describing Poison Ivy rather than simply labeling it with "poison-ivy".

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| anomalous-activity, anonymization, benign, compromised, malicious-activity, attribution | |
| **Vocabulary Value** | **Description** |
| anomalous-activity | An Indicator with this label describes unexpected, or unusual activity that may not necessarily be malicious or indicate compromise. This type of activity may include reconnaissance-like behavior such as port scans or version identification, network behavior anomalies, and asset and/or user behavioral anomalies. |
| anonymization | An Indicator with this label describes suspected anonymization tools or infrastructure (proxy, TOR, VPN, etc.). |
| benign | An Indicator with this label describes activity that is not suspicious or malicious in and of itself, but when combined with other activity may indicate suspicious or malicious behavior. |
| compromised | An Indicator with this label describes assets that are suspected to be compromised. |
| malicious-activity | An Indicator with this label describes patterns of suspected malicious objects and/or activity. |
| attribution | An Indicator with this label describes patterns of behavior that indicate attribution to a particular threat actor or campaign. |

## ​13.9.​ Industry Sector

**Type Name:** industry-sector-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Source
* Threat Actor
* Victim Target

Industry sector is an open vocabulary that describes industrial and commercial sectors. It is intended to be holistic: it has been derived from several other lists and is not limited to "critical infrastructure" sectors.

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| agriculture, aerospace, automotive, communications, construction, defence, education, energy, engineering, entertainment, financial-services, government-national, government-regional, government-local, government-public-services, healthcare, hospitality-leisure, infrastructure, insurance, manufacturing, mining, non-profit, pharmaceuticals, retail, technology, telecommunications, transportation, utilities | |
| **Vocabulary Value** | **Description** |
| agriculture |  |
| aerospace |  |
| automotive |  |
| communications |  |
| construction |  |
| defense |  |
| education |  |
| energy |  |
| engineering |  |
| entertainment |  |
| financial-services |  |
| government-national |  |
| government-regional |  |
| government-local |  |
| government-public-services |  |
| healthcare |  |
| hospitality-leisure |  |
| infrastructure |  |
| insurance |  |
| manufacturing |  |
| mining |  |
| non-profit |  |
| pharmaceuticals |  |
| retail |  |
| technology |  |
| telecommunications |  |
| transportation |  |
| utilities |  |

## ​13.10.​ Malware Label

**Type Name:** malware-label-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Malware

Malware label is an open vocabulary that represents different types and functions of malware. Malware labels are not mutually exclusive: a malware instance can be both spyware and a screen capture tool.

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| adware, backdoor, bot, ddos, dropper, exploit-kit, keylogger, ransomware, remote-access-trojan, resource-exploitation, rogue-antivirus, rootkit, screen-capture, spyware, worm | |
| **Vocabulary Value** | **Description** |
| adware | Any software that is funded by advertising. Adware may also gather sensitive user information from a system. |
| backdoor | A malicious program that allows an attacker to perform actions on a remote system, such as transferring files, acquiring passwords, or executing arbitrary commands [TODO: Ref NIST). |
| bot | A program that resides on an infected system, communicating with and forming part of a botnet. The bot may be implanted by a worm or Trojan, which opens a backdoor. The bot then monitors the backdoor for further instructions. |
| ddos | A tool used to perform a distributed denial of service attack. |
| dropper | A type of trojan that deposits an enclosed payload (generally, other malware) onto the target computer. |
| exploit-kit | A software toolkit to target common vulnerabilities. |
| keylogger | A type of malware that surreptitiously monitors keystrokes and either records them for later retrieval or sends them back to a central collection point. |
| ransomware | A type of malware that encrypts files on a victim's system, demanding payment of ransom in return for the access codes required to unlock files. |
| remote-access-trojan | A remote access Trojan program or RAT, is a Trojan horse capable of controlling a machine through commands issued by a remote attacker. |
| resource-exploitation | A type of malware that steals a system's resources (e.g., CPU cycles), such as a bitcoin miner. |
| rogue-antivirus | A fake security product that demands money to clean phony infections. |
| rootkit | A type of malware that hides its files or processes from normal methods of monitoring in order to conceal its presence and activities. Rootkits can operate at a number of levels, from the application level - simply replacing or adjusting the settings of system software to prevent the display of certain information - through hooking certain functions or inserting modules or drivers into the operating system kernel, to the deeper level of firmware or virtualization rook kits, which are activated before the operating system and thus even harder to detect while the system is running. |
| screen-capture | A type of malware used to capture images from the target systems screen, used for exfiltration and command and control. |
| spyware | Software that gathers information on a user's system without their knowledge and sends it to another party. Spyware is generally used to track activities for the purpose of delivering advertising. |
| worm | A self-replicating, self-contained program that usually executes itself without user intervention. |

​

## ​13.11.​ Pattern Language

**Type Name:** pattern-lang-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Indicator

Pattern Language is an open vocabulary that describes the different types of pattern languages that can be used in a STIX Indicator.

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| cybox, snort, yara | |
| **Vocabulary Value** | **Description** |
| cybox | CybOX Patterning v1.0 [TODO Ref]. cybox is the default value. |
| snort | Snort pattern (any version) |
| yara | Yara pattern (any version) |

## ​13.12.​ Report Label

**Type Name:** report-label-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Report

Report Label is an open vocabulary to describe the primary purpose or subject of a report. For example, a report that contains malware and indicators for that malware should have a report label of malware-report to capture that the malware is the primary purpose. Report labels are not mutually exclusive: a Report can be both a malware report and a tool report.

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| threat-report, attack-pattern-report, campaign-report, indicator-report, malware-report, observed-data-report, threat-actor-report, tool-report, victim-target-report, vulnerability-report | |
| **Vocabulary Value** | **Description** |
| threat-report | Report subject is a broad characterization of a threat across multiple facets. |
| attack-pattern-report | Report subject is a characterization of one or more attack patterns and related information. |
| campaign-report | Report subject is a characterization of one or more campaigns and related information. |
| indicator-report | Report subject is a characterization of one or more indicators and related information. |
| malware-report | Report subject is a characterization of one or more malware instances and related information. |
| observed-data-report | Report subject is a characterization of observed data and related information. |
| threat-actor-report | Report subject is a characterization of one or more threat actors and related information. |
| tool-report | Report subject is a characterization of one or more tools and related information. |
| victim-target-report | Report subject is a characterization of one or more victim targets and related information. |
| vulnerability-report | Report subject is a characterization of one or more vulnerabilities and related information. |

## ​13.13.​ Threat Actor Label

**Type Name:** threat-actor-label-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Threat Actor

Threat actor labels is an open vocabulary used to describe types of threat actors. For example, some threat actors are competitors who try to steal information, while others are activists who act in support of a social or political cause. Actor labels are not mutually exclusive: a threat actor can be both a disgruntled employee and a spy.

[TODO Ref Threat Agent Library, Intel Corporation, September 2007]

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| activist, competitor, crime-syndicate, cyber-warrior, employee-accidental, employee-disgruntled, sensationalist, spy, terrorist, thief | |
| **Vocabulary Value** | **Description** |
| activist | Highly motivated, potentially destructive supporter of a social or political cause.    Activist actions directed towards an organization are often intended to protest and influence the organization’s decisions pertaining to issues such as facility placement, trade and business dealings, or labor or environmental impacts. Their attacks are usually intended to either disrupt the ability produce product or services or damage the company’s image. The activist may act entirely online, or may extend their operations into the cyber realm in addition to physical attacks. Activists are primarily motivated by ideology, which can drive extensive and persistent attacks.    This category includes actors sometimes referred to as anarchists, cyber vandals, extremists, and hacktivists in addition to what are traditionally known as activists. It does not include terrorists, as activist attacks can be severe but generally do not intend the personal injury and loss of life sought by terrorists. |
| competitor | An organization which rivals another in the economic marketplace and competes for the same market share.    The goal of a competitor is to gain an advantage in business with respect to the rival organization it targets. It usually does this by copying intellectual property, trade secrets, acquisition strategies, or other technical or business data from a rival organization with the intention of using the data to bolster its own assets and market position. Highly aggressive competitors may also use disruptive or damaging attacks to slow or block a rival’s progress.    “Competitor” can include vendors and partners, but in this context does not include military adversaries (see the cyber warrior and spy descriptors). The primary motivation for a competitor taking hostile actions is organizational gain. |
| crime-syndicate | An enterprise organized to conduct significant, large-scale criminal activity for profit.    Crime syndicates, also known as organized crime, are generally large, well-resourced groups that operate to create profit from all types of crime. Their activities can be seriously harmful and even extreme in impact, and they may use any combination of physical and cyber techniques to both execute attacks and protect their organization. They are almost entirely motivated by organizational gain to create profit, including cases where they have hired out to political or nationalistic interests to attack on their clients’ behalf. However, they can also act from dominance in establishing local political or social power or in opposing rival syndicates.    As the name implies a crime syndicate is generally a larger, formal organization. Those with similar criminal objectives but working independently or in very small groups generally belong in the Thief category. |
| cyber-warrior | Member of an organization that engages in cyber activities to support active military objectives.    Cyber warriors usually work for organizations affiliated with the military forces of a nation state and work at the direction of that state’s government and military leadership, but may work for a private organization. A cyber warrior typically has access to significant support, resources, training, and tools and is capable of designing and executing very sophisticated and effective campaigns. Using these capabilities, the cyber warrior’s role is to support the organization in active conflicts, either physical or political. Their motivation is primarily dominance, but other motivations such as ideology may come into play.    As in all military organizations, intelligence gathered through espionage is essential to their conflict success and that espionage is often carried out by the same organization. Although affiliated with the cyber warrior, the espionage role is properly called “spy,” even though the individual may actually work in a cyber-war unit and may even take on the cyber warrior role during conflicts. “Cyber warrior” refers only to individuals engaged in active conflicts, including conflicts of the “cold war” type. In cases where the espionage and cyber-war organization are the same, that relationship should be captured as a related-to Relationship from one Threat Actor to another. |
| employee-accidental | A non-hostile employee who unintentionally exposes the organization to harm.    “Employee” in this context includes any person extended internal trust, such as regular employees, contractors, consultants, and temporary workers.    Every employee occasionally makes mistakes, sometimes serious ones. Some risk factors that increase the likelihood of a security-relevant mistake include poor or incomplete training, fatigue, overwork, and distraction. For instance, a new hire may not yet have the knowledge to precisely follow confidentiality protocols, or an experienced worker may be distressed about a relative's illness and forget an important step in a sandbox configuration procedure. In any case, the employee is well-intentioned, and the mistakes are unintentional and possibly even unnoticed by the employee. |
| employee-disgruntled | Current or former employee with intent to harm the organization in retaliation for perceived wrongs.    “Employee” in this context includes any person extended internal trust, such as regular employees, contractors, consultants, and temporary workers.    When the grievances of a disgruntled employee (real or perceived) is severe and the situation escalates, he or she can seek revengeful and harmful retaliation. Disgruntled threat actors can include both employees and former employees, who may have extensive knowledge that can be leveraged when conducting attacks. Often a disgruntled employee acts alone but may join an organization, whether group of similar individuals, a competitor, or criminal organization, if the individual believes that doing so will enable greater harm to the source of his or her anger. A disgruntled person can use cyber or physical means to take any number of actions including sabotage, violence, theft, fraud, espionage, or embarrassing individuals or the organization. |
| sensationalist | Seeks to cause embarrassment and brand damage by exposing sensitive information in a manner designed to cause a public relations crisis.    A Sensationalist may be an individual or small group of people motivated primarily by a need for notoriety. Unlike the Activist, the Sensationalist generally has no political goal, and is not using bad PR to influence the target to change its behavior or business practices. The embarrassment of the target is the end in itself, along with the “15 minutes of fame” that the scandal may bring to the Sensationalists themselves. Any disruption or damage to the target's infrastructure is only important insofar as it adds to negative public perception. |
| spy | Secretly collects the sensitive information of another for use, dissemination, or sale.    While in the broad sense spying, i.e., espionage, is a form of theft, it is recognized as special case and is usually treated far more severely than simple thievery. Many spies are part of a well-resourced intelligence organization and are capable of very sophisticated clandestine operations. However, insiders such as employees or consultants can be just as effective and damaging, even when their activities are largely opportunistic and not part of an overall campaign. This includes employees who leak information they believe is evidence of wrongdoing, or opportunistically taking information when they leave the organization.    In this context, a spy is one who collects sensitive information for the benefit of any economic, industrial, or military espionage objective, in other words the domain or end user is not considered in defining the spy. There can be any number of motivations for spying depending on the individual or organizations involved. |
| terrorist | Uses extreme violence to advance a social or political agenda as well as monetary crimes to support its activities.    “Terrorist” does not have a universally accepted definition and usually depends on regional and situational aspects for identification. In this context it refers to individuals who target noncombatants with extreme violence to send a message of fear far beyond the actual events. They may act independently or as part of a terrorist organization. While terrorist violence requires physical action that action can be facilitated or generated through cyber means, such as by sabotaging critical infrastructure or facility safety systems via cyber manipulation. Terrorist organizations must typically raise much of their operating budget through criminal activity, which often occurs online. Terrorists are also often adept at using and covertly manipulating social media for both recruitment and impact.    The primary motivation for terrorist activity, both violent and monetary, is ideology, which can drive extensive and persistent attacks. Dominance, disgruntlement, and organizational gain are often also present as motivators. |
| thief | Individual who steals items of value for personal financial gain.    A Thief opportunistically attacks wherever it looks like there is easy profit to be made, whether it be from a large company or from another individual. Many kinds of resources can be stolen especially money or other financial assets such as credit card numbers, but also valuables, hardware, business or personal data, intellectual property, or anything else that can be easily sold. Also considered theft are various avenues of extortion, such as ransomware. Theft can be as simple as pocketing an unattended product prototype, and as sophisticated as hacking into a large organization to steal thousands of identities to sell on the black market.    Unlike a spy, who also steals and sells information but for organizational gain, the thief's goal is simple personal financial gain. As defined here, “thief” refers to those acting individually or in very small or informal groups. For sophisticated, organized criminal activity, see the crime syndicate descriptor. |

## ​13.14.​ Threat Actor Role

**Type Name:** threat-actor-role-ov

This vocabulary is currently used in the following SDO(s) in the STIX model:

* Threat Actor

Threat actor roles is an open vocabulary that is used to describe the different roles that a threat actor can play. For example, some threat actors author malware or operate botnets while other actors actually carry out attacks directly.

Threat actor roles are not mutually exclusive: an actor can be both a financial backer for attacks and also direct attacks.

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| agent, director, financial-backer, infrastructure-operator, malware-author, perpetrator | |
| **Vocabulary Value** | **Description** |
| agent | Threat actor is an independent agent. |
| director | Threat actor directs the activities and goals of attacks. |
| financial-backer | Threat actor funds attacks. |
| infrastructure-operator | Threat actor provides attack infrastructure (botnet providers, etc.). |
| malware-author | Threat actor authors malware or other malicious tools. |
| perpetrator | Threat actor performs actual attacks. |

## ​13.15.​ Tool Label

**Type Name:** tool-label-ov

Tool labels describe the categories of tools that can be used to perform attacks.

|  |  |
| --- | --- |
| **Vocabulary Summary** |  |
| denial-of-service, exploitation, network-capture, password-cracking, remote-access, vulnerability-scanning | |
| **Vocabulary Value** | **Description** |
| denial-of-service | A tool used to perform denial of service attacks or distributed denial of service attacks, such as Low Orbit Ion Cannon (LOIC) and DHCPig. |
| exploitation | A tool used to exploit software and systems, such as sqlmap and Metasploit. |
| network-capture | Tools used to capture network traffic, such as Wireshark and Kismet. |
| password-cracking | Tools used to crack password databases, either locally or remotely, such as John the Ripper and NCrack. |
| remote-access | Tools used to access machines remotely, such as VNC and Remote Desktop. |
| vulnerability-scanning | Tools used to scan systems and networks for vulnerabilities, such as NMAP. |

# ​14.​ Customizing STIX

## ​14.1.​ Custom Properties

It is understood that there will be cases where certain information exchanges can be improved by adding properties that are not specified nor reserved in this document; these properties are called **Custom Properties**. This section provides guidance and requirements for how producers can use Custom Properties and how consumers should interpret them in order to extend STIX in an interoperable manner.

### ​14.1.1.​ Requirements

* A STIX Object **MAY** have any number of Custom Properties.
* Custom Property names **MUST** be in ASCII and are limited to characters a-z (lowercase ASCII), 0-9, and underscore (\_).
* Custom Property names **SHOULD** start with “x\_” followed by a source unique identifier (like a domain name), an underscore and then the name. For example: **x\_examplecom\_customfield**.
* Custom Property names **MUST** have a minimum length of 3 ASCII characters.
* Custom Property names **MUST** be no longer than 250 ASCII characters in length.
* Custom Property names that are not prefixed with “x\_” may be used in a future version of the specification for a different meaning. If compatibility with future versions of this specification is required, the “x\_” prefix **MUST** be used.
* Custom Properties **SHOULD** only be used when there is no existing properties defined by the STIX specification that fulfills that need.

A consumer that receives a STIX document with one or more Custom Properties it does not understand **MAY** refuse to process the document further, or silently ignore non-understood properties and continue processing the document.

The reporting and logging of errors originating from the processing of Custom Properties depends heavily on the technology used to transport the STIX document and is therefore not covered in this specification.

*Non-Normative Text*

Producers of STIX documents that contain Custom Properties should be aware of the variability of consumer behavior depending on whether or not the consumer understands the Custom Properties present in a STIX Object. Rules for processing Custom Properties should be well defined and accessible to any consumer that would be reasonably expected to parse them.

### ​14.1.2.​ Examples

{

...,

"x\_acmeinc\_scoring": {

"impact": "high",

"probability": "low"

},

...

}

## ​14.2.​ Custom Objects

It is understood that there will be cases where certain information exchanges can be improved by adding objects that are not specified nor reserved in this document; these objects are called **Custom Objects**. This section provides guidance and requirements for how producers can use Custom Objects and how consumers should interpret them in order to extend STIX in an interoperable manner.

### ​14.2.1.​ Requirements

* Producers **MAY** include any number of Custom Objects in STIX content.
* Custom Objects **MUST** contain the required Common Properties (**id, type, version, modified, created, created\_by\_ref**) and **MAY** contain any optional Common Property (defined in Section TODO).
  + The definitions of these properties are the same as those defined in Common Properties and therefore those fields **MUST NOT** be used to represent the custom properties in the object.
* The **type** field in a Custom Object **MUST** be in ASCII and are limited to the characters a-z (lowercase ASCII), 0-9, and hyphen (-).
* Custom Object names **MUST** have a minimum length of 3 ASCII characters.
* Custom Object names **MUST** be no longer than 250 ASCII characters in length.
* The value of the **type** field in a Custom Object **SHOULD** start with “x-” followed by a source unique identifier (like a domain name), a dash and then the name. For example: x-examplecom-customobject.
* A Custom Object whose name is not prefixed with “x-” may be used in a future version of the specification with a different meaning. Therefore, if compatibility with future versions of this specification is required, the “x-” prefix **MUST** be used.
* The value of the **id** property in a Custom Object **MUST** use the same format as the identifier type, namely, name--uuid
* Custom Objects **SHOULD** only be used when there is no existing STIX Object defined by the STIX specification that fulfills that need.

A consumer that receives a STIX document with one or more Custom Objects that it does not understand **MAY** refuse to process the document further, or silently ignore non-understood objects and continue processing the document.

The reporting and logging of errors originating from the processing of Custom Objects depends heavily on the technology used to transport the STIX document and is therefore not covered in this specification.

*Non-Normative Text*

Producers of STIX documents that contain Custom Objects should be aware of the variability of consumer behavior depending on whether or not the consumer understands the Custom Objects. Rules for processing Custom Objects should be well defined and available to any consumer that would be reasonably expected to parse them.

### ​14.2.2.​ Examples

{

"type": "bundle",

"id": "bundle--f37aa79d-f5f5-4af7-874b-734d32c08c10",

"custom\_objects": [

{

"type": "x-examplecom-customobject",

"id": "x-examplecom-customobject--4527e5de-8572-446a-a57a-706f15467461",

"created": "2016-08-01T00:00:00Z",

"modified": "2016-08-01T00:00:00Z",

"version": 1,

"some\_custom\_stuff": 14,

"other\_custom\_stuff": "hello"

}

]

}​

1. Intel Corp Threat Agent Motivations Feb 2015 [↑](#footnote-ref-0)
2. Intel Corp Threat Agent Library Sept 2007 [↑](#footnote-ref-1)
3. Intel Corp Threat Agent Library Sept 2007 [↑](#footnote-ref-2)
4. Intel Corp Threat Agent Library Sept 2007 [↑](#footnote-ref-3)