STIX™ Version 2.0. Part 2: STIX Objects

Committee Specification Draft ~~01~~02 / Public Review Draft ~~01~~02

~~24 February~~03 May 2017

Specification URIs

This version:

http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part2-stix-objects/stix-v2.0-csprd02-part2-stix-objects.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part2-stix-objects/stix-v2.0-csprd02-part2-stix-objects.html
http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part2-stix-objects/stix-v2.0-csprd02-part2-stix-objects.pdf

Previous version:

http://docs.oasis-open.org/cti/stix/v2.0/csprd01/part2-stix-objects/stix-v2.0-csprd01-part2-stix-objects.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v2.0/csprd01/part2-stix-objects/stix-v2.0-csprd01-part2-stix-objects.html
http://docs.oasis-open.org/cti/stix/v2.0/csprd01/part2-stix-objects/stix-v2.0-csprd01-part2-stix-objects.pdf

~~**Previous version:**~~

~~N/A~~

Latest version:

http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part2-stix-objects.docx (Authoritative)
http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part2-stix-objects.html
http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part2-stix-objects.pdf

Technical Committee:

OASIS Cyber Threat Intelligence (CTI) TC

Chair:

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

Editors:

Rich Piazza (rpiazza@mitre.org), MITRE Corporation
John Wunder (jwunder@mitre.org), MITRE Corporation
Bret Jordan (bret_jordan@symantec.com), Symantec Corp.

Additional artifacts:

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

- *STIX™ Version 2.0. Part 1: STIX Core Concepts*. http://docs.oasis-open.org/cti/stix/v2.0/csprd0~~1~~2/part1-stix-core/stix-v2.0-csprd0~~1~~2-part1-stix-core.html.
- (this document) *STIX™ Version 2.0. Part 2: STIX Objects.* http://docs.oasis-open.org/cti/stix/v2.0/csprd0~~1~~2/part2-stix-objects/stix-v2.0-csprd0~~1~~2-part2-stix-objects.html.
- *STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts.* http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part3-cyber-observable-core/stix-v2.0-csprd02-part3-cyber-observable-core.html.

- *STIX™ Version 2.0. Part 4: Cyber Observable Objects.* http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part4-cyber-observable-objects/stix-v2.0-csprd02-part4-cyber-observable-objects.html.
- *STIX™ Version 2.0. Part 5: STIX Patterning.* http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part5-stix-patterning/stix-v2.0-csprd02-part5-stix-patterning.html.

Related work:

This specification replaces or supersedes:

- *STIX™ Version 1.2.1. Part 1: Overview.* Edited by Sean Barnum, Desiree Beck, Aharon Chernin, and Rich Piazza. Latest version: http://docs.oasis-open.org/cti/stix/v1.2.1/stix-v1.2.1-part1-overview.html.
- *CybOX™ Version 2.1.1. Part 01: Overview.* Edited by Trey Darley, Ivan Kirillov, Rich Piazza, and Desiree Beck. Latest version: http://docs.oasis-open.org/cti/cybox/v2.1.1/cybox-v2.1.1-part01-overview.html.

This specification is related to:

- *TAXII™ Version 2.0.* Edited by ~~Bret Jordan and~~ John Wunder, Mark Davidson, and Bret Jordan. Latest version: http://docs.oasis-open.org/cti/taxii/v2.0/taxii-v2.0.html~~. Work in progress.~~.

Abstract:

Structured Threat Information Expression (STIX™) is a language for expressing cyber threat and observable information. This document defines the set of domain objects and relationship objects that STIX uses to represent cyber threat information.

Status:

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

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

This Committee Specification Public Review Draft is provided under the Non-Assertion Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established. For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the TC's web page (https://www.oasis-open.org/committees/cti/ipr.php).

Note that any machine-readable content (Computer Language Definitions) declared Normative for this Work Product is provided in separate plain text files. In the event of a discrepancy between any such plain text file and display content in the Work Product's prose narrative document(s), the content in the separate plain text file prevails.

Citation format:

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

[STIX-v2.0-Pt2-Objects]

STIX™ Version 2.0. Part 2: STIX Objects. Edited by Rich Piazza, John Wunder, and Bret Jordan. ~~24 February~~03 May 2017. OASIS Committee Specification Draft ~~01~~02 / Public Review Draft ~~01~~02. http://docs.oasis-open.org/cti/stix/v2.0/csprd02/part2-stix-objects/stix-v2.0-csprd02-part2-stix-objects.html. Latest version: http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part2-stix-objects.html.

Notices

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

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

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

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

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

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

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

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

STIX™, CYBOX™, AND TAXII™ (STANDARD OR STANDARDS) AND THEIR COMPONENT PARTS ARE PROVIDED "AS IS" WITHOUT ANY WARRANTY OF ANY KIND, EITHER EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY THAT THESE STANDARDS OR ANY OF THEIR COMPONENT PARTS WILL CONFORM TO SPECIFICATIONS, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR FREEDOM FROM INFRINGEMENT, ANY WARRANTY THAT THE STANDARDS OR THEIR COMPONENT PARTS

Table of Contents

1 Introduction

Structured Threat Information Expression (STIX™) is a language and serialization format used to exchange cyber threat intelligence (CTI). STIX enables organizations to share CTI with one another in a consistent and machine readable manner, allowing security communities to better understand what computer-based attacks they are most likely to see and to anticipate and/or respond to those attacks faster and more effectively. STIX is designed to improve many different capabilities, such as collaborative threat analysis, automated threat exchange, automated detection and response, and 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 properties defined in STIX 1.2.1 (see *STIX™ Version 1.2.1 Part 1: Overview*). The objects chosen for inclusion in STIX 2.0 represent a minimally viable product (MVP) that fulfills basic consumer and producer requirements for CTI sharing. Objects and properties not included in STIX 2.0, but deemed necessary by the community, will be included in future releases.

This document (STIX Objects) uses the concepts introduced in *STIX™ Version 2.0. Part 1: STIX Core Concepts* to define STIX Domain Objects and STIX Relationship Objects.

1.0.1 IPR Policy

This Committee Specification Public Review Draft is provided under the Non-Assertion Mode of the OASIS IPR Policy, the mode chosen when the Technical Committee was established.

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

1.1 Terminology

The key words "**MUST**", "**MUST NOT**", "**REQUIRED**", "**SHALL**", "**SHALL NOT**", "**SHOULD**", "**SHOULD NOT**", "**RECOMMENDED**", "**MAY**", and "**OPTIONAL**" in this document are to be interpreted as described in [RFC2119].

~~**CAPEC** - Common Attack Pattern Enumeration and Classification~~

~~**Consumer** - Any entity that receives STIX content.~~

~~**CTI** - Cyber Threat Intelligence~~

~~**Entity** - Anything that has a separately identifiable existence (e.g., organization, person, group, etc.).~~

~~**IEP** - FIRST (Forum of Incident Response and Security Teams) Information Exchange Policy~~

~~**Instance** - A single occurrence of a STIX object version.~~

~~**MTI** - Mandatory To Implement~~

~~**MVP** - Minimally Viable Product~~

~~**Object Creator** - The entity that created or updated a STIX object (see section 3.3 of).~~

~~**Object Representation** - An instance of an object version that is serialized as STIX.~~

~~**Producer** - Any entity that distributes STIX content, including object creators as well as those passing along existing content.~~

~~**SDO** - STIX Domain Object~~

~~**SRO** - STIX Relationship Object~~

~~**STIX** - Structured Threat Information Expression~~

~~**STIX Content** - STIX documents, including STIX Objects, STIX Objects grouped as bundles, etc.~~

~~**STIX Object** - A STIX Domain Object (SDO) or STIX Relationship Object (SRO)~~

~~**TAXII** - An application layer protocol for the communication of cyber threat information.~~

~~**TLP** - Traffic Light Protocol~~

~~**TTP** - Tactic, technique, or procedure; behaviors and resources that attackers use to carry out their attacks~~

All text is normative except for examples and any text marked non-normative.

1.2 Normative References

[RFC2119] Bradner, S., ~~"~~"Key words for use in RFCs to Indicate Requirement Levels"~~,~~", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997~~.~~, http://www.rfc-editor.org/info/rfc2119~~.~~

1.3 Non-Normative References

[CAPEC] Common Attack Pattern Enumeration and Classification (CAPEC). (2014, Nov. 7). The MITRE Corporation. [Online]. Available: http://capec.mitre.org~~: .~~.

[CVE] Common Vulnerabilities and Exposures (CVE). The MITRE Corporation. [Online]. Available~~: ~~: http://cve.mitre.org.

~~1.41.1 Conventions~~

~~1.4.11.1.1 Naming Conventions~~

~~All type names, property names and literals are in lowercase, except when referencing canonical names defined in another standard (e.g. literal values from an IANA registry). 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.~~

1.4 Naming Requirements

1.4.1 Property Names and String Literals

In the JSON serialization all property names and string literals **MUST** be exactly the same, including case, as the names listed in the property tables in this specification. For example, the SDO common property **`created_by_ref`** must result in the JSON key name "created_by_ref". Properties marked required in the property tables **MUST** be present in the JSON serialization.

1.4.2 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 present in STIX content conforming to this version of the specification.

1.5 Document Conventions

1.5.1 Naming Conventions

All type names, property names and literals are in lowercase, except when referencing canonical names defined in another standard (e.g. literal values from an IANA registry). 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.

~~1.4.3~~1.5.2 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 – `threat-actor`
  - property names are in bold style – **`created_at`**
  - literals (values) are in ~~green~~blue with a ~~green~~blue background – `malicious-activity`
  - All relationship types are string literals, therefore they will also appear in ~~green~~blue with a ~~green~~blue background – `related-to`
- In an object's property table, if a common property is being redefined in some way, then the background is dark grey.
- All examples in this document are expressed in JSON. They are in `Consolas` 9-point font, with straight quotes, black text and a `light ~~blue~~grey background`, and 2-space indentation.
- Parts of the example may be omitted for conciseness and clarity. These omitted parts are denoted with the ellipses (...).

2 STIX™ Domain Objects

This specification defines the set of STIX Domain Objects (SDOs), each of which corresponds to a unique concept commonly represented in CTI. Using SDOs and STIX relationships as building blocks, individuals can create and share broad and comprehensive cyber threat intelligence.

Property information, relationship information, and examples are provided for each SDO defined below. Property information includes common properties as well as properties that are specific to each SDO. Relationship information includes embedded relationships (e.g., **`created_by_ref`**), common relationships (e.g., `related-to`), and SDO-specific relationships. Forward relationships (i.e., relationships *from* the SDO to other SDOs) are fully defined, while reverse relationships (i.e., relationships *to* the SDO from other SDOs) are duplicated for convenience.

Some SDOs are similar and can be grouped together into categories. Attack Pattern, Malware, and Tool can all be considered types of tactics, techniques, and procedures (TTPs): they describe behaviors and resources that attackers use to carry out their attacks. Similarly, Campaign, Intrusion Set, and Threat Actor all describe information about why adversaries carry out attacks and how they organize themselves.

2.1 Attack Pattern

Type Name: `attack-pattern`

Attack Patterns are a type of TTP that describe ways that adversaries attempt to compromise targets. Attack Patterns are used to help categorize attacks, generalize specific attacks to the patterns that they follow, and provide detailed information about how attacks are performed. An example of an attack pattern is "spear phishing": a common type of attack where an attacker sends a carefully crafted e-mail message to a party with the intent of getting them to click a link or open an attachment to deliver malware. Attack Patterns can also be more specific; spear phishing as practiced by a particular threat actor (e.g., they might generally say that the target won a contest) can also be an Attack Pattern.

The Attack Pattern SDO contains textual descriptions of the pattern along with references to externally-defined taxonomies of attacks such as CAPEC [CAPEC]. Relationships from Attack Pattern can be used to relate it to what it targets (Vulnerabilities and Identities) and which tools and malware use it (Tool and Malware).

2.1.1 Properties

Common Properties		
`type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings`		
Attack Pattern Specific Properties		
`name, description, kill_chain_phases`		
Property Name	**Type**	**Description**

type (required)	`string`	The value of ~~this~~ property field **MUST** be `attack-pattern`.
external_references (optional)	`list` of type `external-reference`	A list of external references which refer to non-STIX information. This property **MAY** be used to provide one or more Attack Pattern identifiers, such as a CAPEC ID. When specifying a CAPEC ID, the **source_name** property of the external reference **MUST** be set to `capec` and the **external_id** property **MUST** be formatted as `CAPEC-[id]`.
name (required)	`string`	A name used to identify the Attack Pattern.
description (optional)	`string`	A description that provides more details and context about the Attack Pattern, potentially including its purpose and its key characteristics.
kill_chain_phases (optional)	`list` of type `kill-chain-phase`	The list of Kill Chain Phases for which this Attack Pattern is used.

2.1.2 Relationships

These are the relationships explicitly defined between the Attack Pattern object and other SDOs. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Attack Pattern object by way of the Relationship object. The reverse relationships (relationships "to" the Attack Pattern object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
created_by_ref	`identifier` (of type `identity`)
object_marking_refs	`identifier` (of type `marking-definition`)
Common Relationships	

duplicate-of, derived-from, related-to			
Source	**Relationship Type**	**Target**	**Description**
attack-pattern	targets	identity, vulnerability	This Relationship describes that this Attack Pattern typically targets the type of victims or vulnerability represented by the related Identity or Vulnerability object. For example, a targets Relationship linking an Attack Pattern for SQL injection to an Identity object representing domain administrators means that the form of SQL injection characterized by the Attack Pattern targets domain administrators in order to achieve its objectives. Another example is a Relationship linking an Attack Pattern for SQL injection to a Vulnerability in blogging software means that the particular SQL injection attack exploits that vulnerability.
attack-pattern	uses	malware, tool	This Relationship describes that the related Malware or Tool is used to perform the behavior identified in the Attack Pattern. For example, a uses Relationship linking an Attack Pattern for a distributed denial of service (DDoS) to a Tool for Low Orbit Ion Cannon (LOIC) indicates that the tool can be used to perform those DDoS attacks.
Reverse Relationships			
indicator	indicates	attack-pattern	See forward relationship for definition.
course-of-action	mitigates	attack-pattern	See forward relationship for definition.
campaign, intrusion-set, threat-actor	uses	attack-pattern	See forward relationship for definition.

~~2.1.3~~ Examples

A generic attack pattern for spear phishing, referencing CAPEC

```
{
  "type": "attack-pattern",
  "id": "attack-pattern--0c7b5b88-8ff7-4a4d-aa9d-feb398cd0061",
  "created": "2016-05-12T08:17:27.000Z",
  "modified": "2016-05-12T08:17:27.000Z",
  "name": "Spear Phishing",
  "description": "...",
  "external_references": [
    {
      "source_name": "capec",
      "external_id": "CAPEC-163"
    }
  ]
}
```

A specific attack pattern for a particular form of spear phishing, referencing CAPEC

```
[
  {
    "type": "attack-pattern",
    "id": "attack-pattern--7e33a43e-e34b-40ec-89da-36c9bb2cacd5",
    "created": "2016-05-12T08:17:27.000Z",
    "modified": "2016-05-12T08:17:27.000Z",
    "name": "Spear Phishing as Practiced by Adversary X",
    "description": "A particular form of spear phishing where the attacker claims that the target had won a contest, including personal details, to get them to click on a link.",
    "external_references": [
      {
        "source_name": "capec",
        "id": "CAPEC-163"
      }
    ]
  },
  {
    "type": "relationship",
    "id": "relationship--57b56a43-b8b0-4cba-9deb-34e3e1faed9e",
    "created": "2016-05-12T08:17:27.000Z",
```

```
    "modified": "2016-05-12T08:17:27.000Z",
    "relationship_type": "uses",
    "source_ref": "intrusion-set--0c7e22ad-b099-4dc3-b0df-2ea3f49ae2e6",
    "target_ref": "attack-pattern--7e33a43e-e34b-40ec-89da-36c9bb2cacd5"
  },
  {
    "type": "intrusion-set",
    "id": "intrusion-set--0c7e22ad-b099-4dc3-b0df-2ea3f49ae2e6",
    "created": "2016-05-12T08:17:27.000Z",
    "modified": "2016-05-12T08:17:27.000Z",
    "name": "Adversary X"
  }
]
```

2.2 Campaign

Type Name: `campaign`

A Campaign is a grouping of adversarial behaviors that describes a set of malicious activities or attacks (sometimes called waves) that occur over a period of time against a specific set of targets. Campaigns usually have well defined objectives and may be part of an Intrusion Set.

Campaigns are often attributed to an intrusion set and threat actors. The threat actors may reuse known infrastructure from the intrusion set or may set up new infrastructure specific for conducting that campaign.

Campaigns can be characterized by their objectives and the incidents they cause, people or resources they target, and the resources (infrastructure, intelligence, Malware, Tools, etc.) they use.

For example, a Campaign could be used to describe a crime syndicate's attack using a specific variant of malware and new C2 servers against the executives of ACME Bank during the summer of 2016 in order to gain secret information about an upcoming merger with another bank.

2.2.1 Properties

Common Properties		
type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings		
Campaign Specific Properties		
name, description, aliases, first_seen, last_seen, objective		
Property Name	**Type**	**Description**

type (required)	`string`	The value of this property **MUST** be `campaign`.
name (required)	`string`	A name used to identify the Campaign.
description (optional)	`string`	A description that provides more details and context about the Campaign, potentially including its purpose and its key characteristics.
aliases (optional)	`list` of type `string`	Alternative names used to identify this Campaign
first_seen (optional)	`timestamp`	The time that this Campaign was first seen. This property is a summary property of data from sightings and other data that may or may not be available in STIX. If new sightings are received that are earlier than the first seen timestamp, the object may be updated to account for the new data.
last_seen (optional)	`timestamp`	The time that this Campaign was last seen. This property is a summary property of data from sightings and other data that may or may not be available in STIX. If new sightings are received that are later than the last seen timestamp, the object may be updated to account for the new data.
objective (optional)	`string`	This property defines the Campaign's primary goal, objective, desired outcome, or intended effect — what the Threat Actor hopes to accomplish with this Campaign.

2.2.2 Relationships

These are the relationships explicitly defined between the Campaign object and other objects. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Campaign object by way of the Relationship object. The reverse relationships (relationships "to" the Campaign object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
`created_by_ref`	`identifier` (of type `identity`)
`object_marking_refs`	`identifier` (of type `marking-definition`)

Common Relationships			
`duplicate-of`, `derived-from`, `related-to`			
Source	**Relationship Type**	**Target**	**Description**
`campaign`	`attributed-to`	`intrusion-set`, `threat-actor`	This Relationship describes that the Intrusion Set or Threat Actor that is involved in carrying out the Campaign. For example, an `attributed-to` Relationship from the Glass Gazelle Campaign to the Urban Fowl Threat Actor means that the actor carried out or was involved in some of the activity described by the Campaign.
`campaign`	`targets`	`identity`, `vulnerability`	This Relationship describes that the Campaign uses exploits of the related Vulnerability or targets the type of victims described by the related Identity. For example, a `targets` Relationship from the Glass Gazelle Campaign to a Vulnerability in a blogging platform indicates that attacks performed as part of Glass Gazelle often exploit that Vulnerability. Similarly, a `targets` Relationship from the Glass Gazelle Campaign to a Identity describing the energy sector in the United States means that the Campaign typically carries out attacks against targets in that sector.
`campaign`	`uses`	`attack-pattern`, `malware`, `tool`	This Relationship describes that attacks carried out as part of the Campaign typically use the related Attack Pattern, Malware, or Tool.

			For example, a `uses` Relationship from the Glass Gazelle Campaign to the xInject Malware indicates that xInject is often used during attacks attributed to that Campaign.
Reverse Relationships			
`indicator`	`indicates`	`campaign`	See forward relationship for definition.

~~2.2.3~~ Examples

```
{
  "type": "campaign",
  "id": "campaign--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:00.000Z",
  "modified": "2016-04-06T20:03:00.000Z",
  "name": "Green Group Attacks Against Finance",
  "description": "Campaign by Green Group against a series of targets in the financial services sector."
}
```

2.3 Course of Action

Type Name: `course-of-action`

Note: The Course of Action object in STIX 2.0 is a stub. It is included to support basic use cases (such as sharing prose courses of action) but does not support the ability to represent automated courses of action or contain properties to represent metadata about courses of action. Future STIX 2 releases will expand it to include these capabilities.

A Course of Action is an action taken either to prevent an attack or to respond to an attack that is in progress. It may describe technical, automatable responses (applying patches, reconfiguring firewalls) but can also describe higher level actions like employee training or policy changes. For example, a course of action to mitigate a vulnerability could describe applying the patch that fixes it.

The Course of Action SDO contains a textual description of the action; a reserved **`action`** property also serves as placeholder for future inclusion of machine automatable courses of action. Relationships from the Course of Action can be used to link it to the Vulnerabilities or behaviors (Tool, Malware, Attack Pattern) that it mitigates.

2.3.1 Properties

Common Properties
`type`, `id`, `created_by_ref`, `created`, `modified`, `revoked`, `labels`, `external_references`, `object_marking_refs`, `granular_markings`

Course of Action Specific Properties		
name, description, action		
Property Name	**Type**	**Description**
type (required)	`string`	The value of this property **MUST** be `course-of-action`.
name (required)	`string`	A name used to identify the Course of Action.
description (optional)	`string`	A description that provides more details and context about the Course of Action, potentially including its purpose and its key characteristics.
action (reserved)	`RESERVED`	RESERVED – To capture structured/automated courses of action.

2.3.2 Relationships

These are the relationships explicitly defined between the Course of Action object and other objects. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Course of Action object by way of the Relationship object. The reverse relationships (relationships "to" the Course of Action object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
created_by_ref	`identifier` (of type `identity`)
object_marking_refs	`identifier` (of type `marking-definition`)
Common Relationships	
`duplicate-of`, `derived-from`, `related-to`	

Source	Relationship Type	Target	Description
course-of-action	mitigates	attack-pattern, malware, tool, vulnerability	This Relationship describes that the Course of Action can mitigate the related Attack Pattern, Malware, Vulnerability, or Tool. For example, a mitigates Relationship from a Course of Action object to a Malware object indicates that the course of action mitigates the impact of that malware.
Reverse Relationships			
—	—	—	—

~~2.3.3~~ Examples

```
[
  {
    {
      "type": "course-of-action",
      "id": "course-of-action--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
      "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
      "created": "2016-04-06T20:03:48.000Z",
      "modified": "2016-04-06T20:03:48.000Z",
      "name": "Add TCP port 80 Filter Rule to the existing Block UDP 1434 Filter",
      "description": "This is how to add a filter rule to block inbound access to TCP port 80 to the existing UDP 1434 filter ..."
    },
    {
    },
    {
      "type": "relationship",
      "id": "relationship--44298a74-ba52-4f0c-87a3-1824e67d7fad",
      "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
      "created": "2016-04-06T20:06:3707:10.000Z",
      "modified": "2016-04-06T20:06:3707:10.000Z",
      "relationship_type": "mitigates",
      "source_ref": "course-of-action--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
      "target_ref": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b","
      "relationship_type": "mitigates"
    },
    {
```

```
  },
  {
    "type": "malware",
    "id": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b",
    "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
    "created": "2016-04-06T20:07:09.000Z",
    "modified": "2016-04-06T20:07:09.000Z",
    "relationship_type": "name": "Poison Ivy"
  }
]
```

2.4 Identity

Type Name: `identity`

Identities can represent actual individuals, organizations, or groups (e.g., ACME, Inc.) as well as classes of individuals, organizations, or groups (e.g., the finance sector).

The Identity SDO can capture basic identifying information, contact information, and the sectors that the Identity belongs to. Identity is used in STIX to represent, among other things, targets of attacks, information sources, object creators, and threat actor identities.

2.4.1 Properties

Common Properties		
`type`, `id`, `created_by_ref`, `created`, `modified`, `revoked`, `labels`, `external_references`, `object_marking_refs`, `granular_markings`		
Identity Specific Properties		
`name`, `description`, `identity_class`, `sectors`, `contact_information`		
Property Name	**Type**	**Description**
`type` (required)	`string`	The value of this property **MUST** be `identity`.
`labels` (optional)	`list` of type `string`	The list of roles that this Identity performs (e.g., CEO, Domain Administrators, Doctors, Hospital, or Retailer). No open vocabulary is yet defined for this property.

name (required)	`string`	The name of this Identity. When referring to a specific entity (e.g., an individual or organization), this property **SHOULD** contain the canonical name of the specific entity.
description (optional)	`string`	A description that provides more details and context about the Identity, potentially including its purpose and its key characteristics.
identity_class (required)	`open-vocab`	The type of entity that this Identity describes, e.g., an individual or organization. This is an open vocabulary and the values **SHOULD** come from the `identity-class-ov` vocabulary.
sectors (optional)	`list` of type `open-vocab`	The list of industry sectors that this Identity belongs to. This is an open vocabulary and values **SHOULD** come from the `industry-sector-ov` vocabulary.
contact_information (optional)	`string`	The contact information (e-mail, phone number, etc.) for this Identity. No format for this information is currently defined by this specification.

2.4.2 Relationships

There is an embedded relationship to Identity in all STIX Objects called **created_by_ref** that is inherited from the Common Properties. This property links each object with the Identity of the organization or individual that created the object.

These are the relationships explicitly defined between the Identity object and other objects. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Identity object by way of the Relationship object. None are defined for the Identity object. The reverse relationships (relationships "to" the Identity object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
created_by_ref	identifier (of type identity)
object_marking_refs	identifier (of type marking-definition)

Common Relationships			
duplicate-of, derived-from, related-to			
Source	**Relationship Type**	**Target**	**Description**
—	—	—	—
Reverse Relationships			
attack-pattern, campaign, intrusion-set, malware, threat-actor, tool	targets	identity	See forward relationship for definition.
threat-actor	attributed-to, impersonates	identity	See forward relationship for definition.

~~2.4.3~~ Examples

An Identity for an individual named John Smith

```
{
  "type": "identity",
  ...,
-  "id": "identity--023d105b-752e-4e3c-941c-7d3f3cb15e9e",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:00.000Z",
  "modified": "2016-04-06T20:03:00.000Z",
  "name": "John Smith",
  "identity_class": "individual",
  ...
}
```

An Identity for a company named ACME Widget, Inc.

```
{
  "type": "identity",
  ...,
  "id": "identity--e5f1b90a-d9b6-40ab-81a9-8a29df4b6b65",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:00.000Z",
  "modified": "2016-04-06T20:03:00.000Z",
  "name": "ACME Widget, Inc.",
  "identity_class": "organization",
  ...
}
```

2.5 Indicator

Type Name: `indicator`

Indicators contain a pattern that can be used to detect suspicious or malicious cyber activity. For example, an Indicator may be used to represent a set of malicious domains and use the STIX Patterning Language (*STIX™ Version 2.0. Part 5: STIX Patterning*) to specify these domains.

The Indicator SDO contains a simple textual description, the Kill Chain Phases that it detects behavior in, a time window for when the Indicator is valid or useful, and a required **`pattern`** property to capture a structured detection pattern. Conforming STIX implementations **MUST** support the STIX Patterning Language as defined in *STIX™ Version 2.0. Part 5: STIX Patterning*. While each structured pattern language has different syntax and potentially different semantics, in general an Indicator is considered to have "matched" (or been "sighted") when the conditions specified in the structured pattern are satisfied in whatever context they are evaluated in.

Relationships from the Indicator can describe the malicious or suspicious behavior that it directly detects (Malware, Tool, and Attack Pattern) as well as the Campaigns, Intrusion Sets, and Threat Actors that it might indicate the presence of.

2.5.1 Properties

Common Properties		
`type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings`		
Indicator Specific Properties		
`name, description, pattern, valid_from, valid_until, kill_chain_phases`		
Property Name	**Type**	**Description**
`type` (required)	`string`	The value of this property **MUST** be

		`indicator`.
`labels` (required)	`list` of type `open-vocab`	This property is an Open Vocabulary that specifies the type of indicator. This is an open vocabulary and values **SHOULD** come from the `indicator-label-ov` vocabulary.
`name` (optional)	`string`	A name used to identify the Indicator.
`description` (optional)	`string`	A description that provides more details and context about the Indicator, potentially including its purpose and its key characteristics.
`pattern` (required)	`string`	The detection pattern for this Indicator is a STIX Pattern as specified in *STIX™ Version 2.0. Part 5: STIX Patterning*.
`valid_from` (required)	`timestamp`	The time from which this Indicator should be considered valuable intelligence.
`valid_until` (optional)	`timestamp`	The time at which this Indicator should no longer be considered valuable intelligence. If the **`valid_until`** property is omitted, then there is no constraint on the latest time for which the Indicator should be used.
`kill_chain_phases` (optional)	`list` of type `kill-chain-phase`	The kill chain phase(s) to which this Indicator corresponds.

2.5.2 Relationships

These are the relationships explicitly defined between the Indicator object and other objects. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Indicator object by way of the Relationship object. The reverse relationships (relationships "to" the Indicator object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
created_by_ref	identifier (of type identity)
object_marking_refs	identifier (of type marking-definition)

Common Relationships			
duplicate-of, derived-from, related-to			
Source	**Relationship Type**	**Target**	**Description**
indicator	indicates	attack-pattern, campaign, intrusion-set, malware, threat-actor, tool	This Relationship describes that the Indicator can detect evidence of the related Campaign, Intrusion Set, or Threat Actor. This evidence may not be direct: for example, the Indicator may detect secondary evidence of the Campaign, such as malware or behavior commonly used by that Campaign. For example, an indicates Relationship from an Indicator to a Campaign object representing Glass Gazelle means that the Indicator is capable of detecting evidence of Glass Gazelle, such as command and control IPs commonly used by that Campaign.
Reverse Relationships			
—	—	—	—

~~2.5.3~~ Examples

Indicator itself, with context

```
[
  {
    "type": "indicator",
    "id": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
    "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
    "created": "2016-04-06T20:03:48.000Z",
    "modified": "2016-04-06T20:03:48.000Z",
    "labels": ["malicious-activity"],
```

```
    "name": "Poison Ivy Malware",
    "description": "This file is part of Poison Ivy",
    "pattern": "[ file:hashes.MD5 = '3773a88f65a5e780c8dff9cdc3a056f3' 'SHA-256' =
'4bac27393bdd9777ce02453256c5577cd02275510b2227f473d03f533924f877' ]",
    "valid_from": "2016-01-01T00:00:00Z"
  },
  {
    "type": "relationship",
    "id": "relationship--44298a74-ba52-4f0c-87a3-1824e67d7fad",
    "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
    "created": "2016-04-06T20:06:37.000Z",
    "modified": "2016-04-06T20:06:37.000Z",
    "source_ref": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
    "target_ref": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b",
    "relationship_type": "indicates"
  },
  {
    "type": "malware",
    "id": "malware--31b940d4-6f7f-459a-80ea-9c1f17b5891b",
    "created": "2016-04-06T20:07:09.000Z",
    "modified": "2016-04-06T20:07:09.000Z",
    "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
    "name": "Poison Ivy"
  }
]
```

2.6 Intrusion Set

Type Name: `intrusion-set`

An Intrusion Set is a grouped set of adversarial behaviors and resources with common properties that is believed to be orchestrated by a single organization. An Intrusion Set may capture multiple Campaigns or other activities that are all tied together by shared attributes indicating a common known or unknown Threat Actor. New activity can be attributed to an Intrusion Set even if the Threat Actors behind the attack are not known. Threat Actors can move from supporting one Intrusion Set to supporting another, or they may support multiple Intrusion Sets.

Where a Campaign is a set of attacks over a period of time against a specific set of targets to achieve some objective, an Intrusion Set is the entire attack package and may be used over a very long period of time in multiple Campaigns to achieve potentially multiple purposes.

While sometimes an Intrusion Set is not active, or changes focus, it is usually difficult to know if it has truly disappeared or ended. Analysts may have varying level of fidelity on attributing an Intrusion Set back to

Threat Actors and may be able to only attribute it back to a nation state or perhaps back to an organization within that nation state.

2.6.1 Properties

Common Properties
type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings
Intrusion Set Specific Properties
name, description, aliases, first_seen, last_seen, goals, resource_level, primary_motivation, secondary_motivations

Property Name	Type	Description
type (required)	`string`	The value of this property **MUST** be `intrusion-set`.
name (required)	`string`	A name used to identify this Intrusion Set.
description (optional)	`string`	A description that provides more details and context about the Intrusion Set, potentially including its purpose and its key characteristics.
aliases (optional)	`list` of type `string`	Alternative names used to identify this Intrusion Set.
first_seen (optional)	`timestamp`	The time that this Intrusion Set was first seen. This property is a summary property of data from sightings and other data that may or may not be available in STIX. If new sightings are received that are earlier than the first seen timestamp, the object may be updated to account for the new data.
last_seen (optional)	`timestamp`	The time that this Intrusion Set was last seen.

		This property is a summary property of data from sightings and other data that may or may not be available in STIX. If new sightings are received that are later than the last seen timestamp, the object may be updated to account for the new data.
goals (optional)	`list` of type `string`	The high level goals of this Intrusion Set, namely, *what* are they trying to do. For example, they may be motivated by personal gain, but their goal is to steal credit card numbers. To do this, they may execute specific Campaigns that have detailed objectives like compromising point of sale systems at a large retailer. Another example: to gain information about latest merger and IPO information from ACME Bank.
resource_level (optional)	`open-vocab`	This defines the organizational level at which this Intrusion Set typically works, which in turn determines the resources available to this Intrusion Set for use in an attack. This is an open vocabulary and values **SHOULD** come from the `attack-resource-level-ov` vocabulary.
primary_motivation (optional)	`open-vocab`	The primary reason, motivation, or purpose behind this Intrusion Set. The motivation is *why* the Intrusion Set wishes to achieve the goal (what they are trying to achieve). For example, an Intrusion Set with a goal to disrupt the finance sector in a country might be motivated by ideological hatred of capitalism. This is an open vocabulary and values **SHOULD** come from the `attack-motivation-ov` vocabulary.
secondary_motivations (optional)	`list` of type `open-vocab`	The secondary reasons, motivations, or purposes behind this Intrusion Set. These motivations can exist as an equal or near-equal cause to the

		primary motivation. However, it does not replace or necessarily magnify the primary motivation, but it might indicate additional context. This is an open vocabulary and values **SHOULD** come from the `attack-motivation-ov` vocabulary.

2.6.2 Relationships

These are the relationships explicitly defined between the Intrusion Set object and other objects. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Intrusion Set object by way of the Relationship object. The reverse relationships (relationships "to" the Intrusion Set object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
`created_by_ref`	`identifier` (of type `identity`)
`object_marking_refs`	`identifier` (of type `marking-definition`)

Common Relationships			
`duplicate-of`, `derived-from`, `related-to`			
Source	**Relationship Type**	**Target**	**Description**
`intrusion-set`	`attributed-to`	`threat-actor`	This Relationship describes that the related Threat Actor is involved in carrying out the Intrusion Set. For example, an `attributed-to` Relationship from the Red Orca Intrusion Set to the Urban Fowl Threat Actor means that the actor carried out or was involved in some of the activity described by the Intrusion Set.
`intrusion-set`	`targets`	`identity`, `vulnerability`	This Relationship describes that the Intrusion Set uses exploits of the related Vulnerability or targets the type of victims described by the related Identity.

			For example, a `targets` Relationship from the Red Orca Intrusion Set to a Vulnerability in a blogging platform indicates that attacks performed as part of Red Orca often exploit that Vulnerability. Similarly, a `targets` Relationship from the Red Orca Intrusion Set to an Identity describing the energy sector in the United States means that the Intrusion Set typically carries out attacks against targets in that sector.
`intrusion-set`	`uses`	`attack-pattern`, `malware`, `tool`	This Relationship describes that attacks carried out as part of the Intrusion Set typically use the related Attack Pattern, Malware, or Tool. For example, a `uses` Relationship from the Red Orca Intrusion Set to the xInject Malware indicates that xInject is often used during attacks attributed to that Intrusion Set.
Reverse Relationships			
`campaign`	`attributed-to`	`intrusion-set`	See forward relationship for definition.
`indicator`	`indicates`	`intrusion-set`	See forward relationship for definition.

~~2.6.3~~ Examples

```
{
  "type": "intrusion-set",
  "id": "intrusion-set--4e78f46f-a023-4e5f-bc24-71b3ca22ec29",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:48.000Z",
  "modified": "2016-04-06T20:03:48.000Z",
  "name": "Bobcat Breakin",
  "description": "Incidents usually feature a shared TTP of a bobcat being released within the building containing network access, scaring users to leave their computers without locking them first. Still determining where the threat actors are getting the bobcats.",
  "aliases": ["Zookeeper"],
  "goals": ["acquisition-theft", "harassment", "damage"]
}
```

2.7 Malware

Type Name: `malware`

Note: The Malware object in STIX 2.0 is a stub. It is included to support basic use cases but is likely not useful for actual malware analysis or for including even simple malware instance data. Future versions of STIX 2 will expand it to include these capabilities.

Malware is a type of TTP that is also known as malicious code and malicious software, and refers to a program that is inserted into a system, usually covertly, with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system (OS) or of otherwise annoying or disrupting the victim. Malware such as viruses and worms are usually designed to perform these nefarious functions in such a way that users are unaware of them, at least initially.[1]

The Malware SDO characterizes, identifies, and categorizes malware samples and families via a text **`description`** property. This provides detailed information about how the malware works and what it does. Relationships from Malware can capture what the malware targets (Vulnerability and Identity) and link it to another Malware SDO that it is a variant of.

2.7.1 Properties

Common Properties		
`type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings`		
Malware Specific Properties		
`name, description, kill_chain_phases`		
Property Name	**Type**	**Description**
`type` (required)	`string`	The value of this property **MUST** be `malware`.
`labels` (required)	`list` of type `open-vocab`	The type of malware being described. This is an open vocabulary and values **SHOULD** come from the `malware-label-ov` vocabulary.

[1] NIST SP 800-83. http://csrc.nist.gov/publications/nistpubs/800-83/SP800-83.pdf.

name (required)	`string`	A name used to identify the Malware sample.
description (optional)	`string`	A description that provides more details and context about the Malware, potentially including its purpose and its key characteristics.
kill_chain_phases (optional)	`list` of type `kill-chain-phase`	The list of Kill Chain Phases for which this Malware can be used.

2.7.2 Relationships

These are the relationships explicitly defined between the Malware object and other objects. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Malware object by way of the Relationship object. The reverse relationships (relationships "to" the Malware object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
created_by_ref	`identifier` (of type `identity`)
object_marking_refs	`identifier` (of type `marking-definition`)

Common Relationships			
`duplicate-of`, `derived-from`, `related-to`			
Source	**Relationship Type**	**Target**	**Description**
`malware`	`targets`	`identity`, `vulnerability`	This Relationship documents that this Malware is being used to target this Identity or exploit the Vulnerability. For example, a `targets` Relationship linking a Malware representing a downloader to a Vulnerability for CVE-

			2016-0001 means that the malware exploits that vulnerability. Similarly, a `targets` Relationship linking a Malware representing a downloader to an Identity representing the energy sector means that downloader is typically used against targets in the energy sector.
`malware`	`uses`	`tool`	This Relationship documents that this Malware uses the related tool to perform its functions.
`malware`	`variant-of`	`malware`	This Relationship is used to document that one piece of Malware is a variant of another piece of Malware. For example, TorrentLocker is a variant of CryptoLocker.
Reverse Relationships			
`indicator`	`indicates`	`malware`	See forward relationship for definition.
`course-of-action`	`mitigates`	`malware`	See forward relationship for definition.
`attack-pattern`, `campaign`, `intrusion-set`, `threat-actor`	`uses`	`malware`	See forward relationship for definition.

~~2.7.3~~ Examples

```
{
  "type": "malware",
  "id": "malware--0c7b5b88-8ff7-4a4d-aa9d-feb398cd0061",
  "created": "2016-05-12T08:17:27.000Z",
  "modified": "2016-05-12T08:17:27.000Z",
  "name": "Cryptolocker",
  "description": "...",
  "labels": ["ransomware"]
}
```

2.8 Observed Data

Type Name: `observed-data`

Observed Data conveys information that was observed on systems and networks using the Cyber Observable specification defined in parts 3 and 4 of this specification. For example, Observed Data can capture the observation of an IP address, a network connection, a file, or a registry key. Observed Data is not an intelligence assertion, it is simply information: this file was seen, without any context for what it means.

Observed Data captures both a single observation of a single entity (file, network connection) as well as the aggregation of multiple observations of an entity. When the **number_observed** property is `1` the Observed Data is of a single entity. When the **number_observed** property is greater than `1`, the observed data consists of several instances of an entity collected over the time window specified by the **first_observed** and **last_observed** properties. When used to collect aggregate data, it is likely that some fields in the Cyber Observable Object (e.g., timestamp fields) will be omitted because they would differ for each of the individual observations.

Observed Data may be used by itself (without relationships) to convey raw data collected from network and host-based detection tools. A firewall could emit a single Observed Data instance containing a single Network Traffic object for each connection it sees. The firewall could also aggregate data and instead send out an Observed Data instance every ten minutes with an IP address and an appropriate **number_observed** value to indicate the number of times that IP address was observed in that window.

Observed Data may also be related to other SDOs to represent raw data that is relevant to those objects. The Sighting object, which captures the sighting of an Indicator, Malware, or other SDO, uses Observed Data to represent the raw information that led to the creation of the Sighting (e.g., what was actually seen that suggested that a particular instance of malware was active).

2.8.1 Properties

Common Properties		
type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings		
Observed Data Specific Properties		
first_observed, last_observed, number_observed, objects		
Property Name	**Type**	**Description**
type (required)	`string`	The value of this property **MUST** be `observed-data`.
first_observed (required)	`timestamp`	The beginning of the time window during which the data was observed.
last_observed (required)	`timestamp`	The end of the time window during which the data was observed.

number_observed (required)	integer	The number of times the data represented in the **objects** property was observed. This **MUST** be an integer between 1 and 999,999,999 inclusive. If the **number_observed** property is greater than 1, the data contained in the **objects** property was observed multiple times. In these cases, object creators **MAY** omit properties of the Cyber Observable object (such as timestamps) that are specific to a single instance of that observed data.
objects (required)	observable-objects	A dictionary of Cyber Observable Objects representing the observation. The dictionary **MUST** contain at least one object. The observable-objects type is defined in *STIX™ Version 2.0. Part 3: Cyber Observable Core Concepts*. The Cyber Observable content **MAY** include multiple objects if those objects are related as part of a single observation. Multiple objects not related to each other via Cyber Observable Relationships **MUST NOT** be contained within the same Observed Data instance. For example, a Network Traffic object and two IPv4 Address objects related via the **src_ref** and **dst_ref** properties can be contained in the same Observed Data because they are all related and used to characterize that single entity. Two unrelated IPv4 address objects that just happened to be observed at the same time, however, must be represented in separate Observed Data instances.

2.8.2 Relationships

There are no relationships explicitly defined between the Observed Data 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.

In addition to the relationships created using the generic Relationship object, Observed Data is also a direct target of the Sighting SRO. Sightings represent a relationship between some intelligence entity that was seen (e.g., an Indicator or Malware instance), where it was seen, and what evidence was actually seen. The evidence (or raw data) in that relationship is captured as Observed Data.

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

Embedded Relationships	
created_by_ref	`identifier` (of type `identity`)
object_marking_refs	`identifier` (of type `marking-definition`)
Common Relationships	
`duplicate-of`, `derived-from`, `related-to`	

Source	Name	Target	Description
—	—	—	—

~~2.8.3~~ Examples

Observed Data of a File object

```
{
  "type": "observed-data",
  "id": "observed-data--b67d30ff-02ac-498a-92f9-32f845f448cf",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T19:58:16.000Z",
  "modified": "2016-04-06T19:58:16.000Z",
  "first_observed": "2015-12-21T19:00:00Z",
  "last_observed": "2015-12-21T19:00:00Z",
  "number_observed": 50,
  "objects": {
    "0": {
      "type": "file",
      ...
    }
    ...
  }
}
```

2.9 Report

Type Name: `report`

Reports are collections of threat intelligence focused on one or more topics, such as a description of a threat actor, malware, or attack technique, including context and related details. They are used to group related threat intelligence together so that it can be published as a comprehensive cyber threat story.

The Report SDO contains a list of references to SDOs and SROs (the CTI objects included in the report) along with a textual description and the name of the report.

For example, a threat report produced by ACME Defense Corp. discussing the Glass Gazelle campaign should be represented using Report. The Report itself would contain the narrative of the report while the Campaign SDO and any related SDOs (e.g., Indicators for the Campaign, Malware it uses, and the associated Relationships) would be referenced in the report contents.

2.9.1 Properties

Common Properties		
`type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings`		
Report Specific Properties		
`name, description, published, object_refs`		
Property Name	**Type**	**Description**
`type` (required)	`string`	The value of this property **MUST** be `report`.
`labels` (required)	`list` of type `open-vocab`	This property is an Open Vocabulary that specifies the primary subject of this report. This is an open vocabulary and values **SHOULD** come from the `report-label-ov` vocabulary.
`name` (required)	`string`	A name used to identify the Report.
`description` (optional)	`string`	A description that provides more details and context about the Report, potentially including its purpose and its key characteristics.

published (required)	`timestamp`	The date that this Report object was officially published by the creator of this report. The publication date (public release, legal release, etc.) may be different than the date the report was created or shared internally (the date in the **created** property).
object_refs (required)	`list` of type `identifier`	Specifies the STIX Objects that are referred to by this Report.

2.9.2 Relationships

There are no relationships explicitly defined between the Report 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	`identifier` (of type `identity`)
object_marking_refs	`identifier` (of type `marking-definition`)
object_refs	`list` of type `identifier` (of STIX Object or `marking-definition` type)
Common Relationships	
`duplicate-of`, `derived-from`, `related-to`	

~~2.9.3~~ Examples

A standalone Report; the consumer may or may not already have access to the referenced STIX Objects.

```
{
  "type": "report",
  "id": "report--84e4d88f-44ea-4bcd-bbf3-b2c1c320bcb3",
  "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
  "created": "2015-12-21T19:59:11.000Z",
  "modified": "2015-12-21T19:59:11.000Z",
```

```
  "name": "The Black Vine Cyberespionage Group",
  "description": "A simple report with an indicator and campaign",
  "published": "2016-01-201T17:00:00Z00.000Z",
  "labels": ["campaign"],
  "object_refs": [
    "indicator--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
    "campaign--83422c77-904c-4dc1-aff5-5c38f3a2c55c",
    "relationship--f82356ae-fe6c-437c-9c24-6b64314ae68a"
  ]
}
```

A Bundle with a Report and the STIX Objects that are referred to by the Report

```
{
  "type": "bundle",
  "id": "bundle--44af6c39-c09b-49c5-9de2-394224b04982",
  "objects": [
    {
      "type": "identity",
      "id": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
      ...,
      ...,
      "name": "Acme Cybersecurity Solutions"
    },
    {
      },
    {
      "type": "report",
      "id": "report--84e4d88f-44ea-4bcd-bbf3-b2c1c320bcbd",
      "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
      "created": "2015-12-21T19:59:11.000Z",
      "modified": "2016-05-21T19:59:11.000Z",
      "name": "The Black Vine Cyberespionage Group",
      "description": "A simple report with an indicator and campaign",
      "published": "2016-01-201T17:00:00Z",
      "labels": ["campaign"],
      "object_refs": [
        "indicator--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
        "campaign--83422c77-904c-4dc1-aff5-5c38f3a2c55c",
        "relationship--f82356ae-fe6c-437c-9c24-6b64314ae68a"
      ]
    },
    {
      ]
    },
    {
```

```
      "type": "indicator",
      "id": "indicator--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
      "created": "2015-12-21T19:59:17.000Z",
      "modified": "2016-05-21T19:59:17.000Z",
      "name": "Some indicator",
      "labels": ["malicious-activity"],
      "pattern": "[ file:hashes.MD5 = '3773a88f65a5e780c8dff9cdc3a056f3' ]",
      "valid_from": "2015-12-21T19:59:17Z",
      "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283"
    },
    {
    },
    {
      "type": "campaign",
      "id": "campaign--83422c77-904c-4dc1-aff5-5c38f3a2c55c",
      "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
      "created": "2015-12-21T19:59:17.000Z",
      "modified": "2016-05-21T19:59:17.000Z",
      "name": "Some Campaign"
    },
    {
    },
    {
      "id": "relationship--f82356ae-fe6c-437c-9c24-6b64314ae68a",
      "type": "relationship",
      "created_by_ref": "identity--a463ffb3-1bd9-4d94-b02d-74e4f1658283",
      "created": "2015-12-21T19:59:17.000Z",
      "modified": "2015-12-21T19:59:17.000Z",
      "source_ref": "indicator--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
      "target_ref": "campaign--26ffb872-1dd9-446e-b6f5-d58527e5b5d2",
      "name": "indicates"
    }
    }
  ]
}
```

2.10 Threat Actor

Type Name: `threat-actor`

Threat Actors are actual individuals, groups, or organizations believed to be operating with malicious intent. A Threat Actor is not an Intrusion Set but may support or be affiliated with various Intrusion Sets, groups, or organizations over time.

Threat Actors leverage their resources, and possibly the resources of an Intrusion Set, to conduct attacks and run Campaigns against targets.

Threat Actors can be characterized by their motives, capabilities, goals, sophistication level, past activities, resources they have access to, and their role in the organization.

2.10.1 Properties

Common Properties
type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings
Threat Actor Specific Properties
name, description, aliases, roles, goals, sophistication, resource_level, primary_motivation, secondary_motivations, personal_motivations

Property Name	Type	Description
type (required)	`string`	The value of this property **MUST** be `threat-actor`.
labels (required)	`list` of type `open-vocab`	This property specifies the type of Threat Actor. This is an open vocabulary and values **SHOULD** come from the `threat-actor-label-ov` vocabulary.
name (required)	`string`	A name used to identify this Threat Actor or Threat Actor group.
description (optional)	`string`	A description that provides more details and context about the Threat Actor, potentially including its purpose and its key characteristics.
aliases (optional)	`list` of type `string`	A list of other names that this Threat Actor is believed to use.
roles (optional)	`list` of type `open-vocab`	A list of roles the Threat Actor plays. This is an open vocabulary and the values **SHOULD** come from the `threat-actor-role-ov` vocabulary.
goals (optional)	`list` of type `string`	The high level goals of this Threat Actor, namely, *what* are they trying to do. For example, they may be motivated by personal gain, but their goal is to

		steal credit card numbers. To do this, they may execute specific Campaigns that have detailed objectives like compromising point of sale systems at a large retailer.
sophistication (optional)	`open-vocab`	The skill, specific knowledge, special training, or expertise a Threat Actor must have to perform the attack. This is an open vocabulary and values **SHOULD** come from the `threat-actor-sophistication-ov` vocabulary.
resource_level (optional)	`open-vocab`	This defines the organizational level at which this Threat Actor typically works, which in turn determines the resources available to this Threat Actor for use in an attack. This attribute is linked to the **sophistication** property — a specific resource level implies that the Threat Actor has access to at least a specific sophistication level. This is an open vocabulary and values **SHOULD** come from the `attack-resource-level-ov` vocabulary.
primary_motivation (optional)	`open-vocab`	The primary reason, motivation, or purpose behind this Threat Actor. The motivation is *why* the Threat Actor wishes to achieve the goal (what they are trying to achieve). For example, a Threat Actor with a goal to disrupt the finance sector in a country might be motivated by ideological hatred of capitalism. This is an open vocabulary and values **SHOULD** come from the `attack-motivation-ov` vocabulary.
secondary_motivations (optional)	`list` of type `open-vocab`	The secondary reasons, motivations, or purposes behind this Threat Actor. These motivations can exist as an equal or near-equal cause to the primary motivation. However, it does not replace or necessarily magnify the primary motivation, but it might indicate additional context.

		This is an open vocabulary and values **SHOULD** come from the `attack-motivation-ov` vocabulary.
`personal_motivations` (optional)	`list` of type `open-vocab`	The personal reasons, motivations, or purposes of the Threat Actor regardless of organizational goals. Personal motivation, which is independent of the organization's goals, describes what impels an individual to carry out an attack. Personal motivation may align with the organization's motivation—as is common with activists—but more often it supports personal goals. For example, an individual analyst may join a Data Miner corporation because his or her skills may align with the corporation's objectives. But the analyst most likely performs his or her daily work toward those objectives for personal reward in the form of a paycheck. The motivation of personal reward may be even stronger for Threat Actors who commit illegal acts, as it is more difficult for someone to cross that line purely for altruistic reasons. This is an open vocabulary and values **SHOULD** come from the `attack-motivation-ov` vocabulary.

2.10.2 Relationships

These are the relationships explicitly defined between the Threat Actor object and other objects. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Threat Actor object by way of the Relationship object. The reverse relationships (relationships "to" the Threat Actor object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
`created_by_ref`	`identifier` (of type `identity`)
`object_marking_refs`	`identifier` (of type `marking-definition`)

Common Relationships			
duplicate-of, derived-from, related-to			
Source	**Relationship Type**	**Target**	**Description**
threat-actor	attributed-to	identity	This Relationship describes that the Threat Actor's real identity is the related Identity. For example, an attributed-to Relationship from the jay-sm17h Threat Actor to the John Smith Identity means that the actor known as jay-sm17h is John Smith.
threat-actor	impersonates	identity	This Relationship describes that the Threat Actor impersonates the related Identity. For example, an impersonates Relationship from the gh0st Threat Actor to the ACME Corp. Identity means that the actor known as gh0st impersonates ACME Corp.
threat-actor	targets	identity, vulnerability	This Relationship describes that the Threat Actor uses exploits of the related Vulnerability or targets the type of victims described by the related Identity. For example, a targets Relationship from the jay-sm17h Threat Actor to a Vulnerability in a blogging platform indicates that attacks performed by John Smith often exploit that Vulnerability. Similarly, a targets Relationship from the jay-sm17h Threat Actor to an Identity describing the energy sector in the United States means that John Smith often carries out attacks against targets in that sector.

threat-actor	uses	attack-pattern, malware, tool	This Relationship describes that attacks carried out as part of the Threat Actor typically use the related Attack Pattern, Malware, or Tool. For example, a uses Relationship from the jay-sm17h Threat Actor to the xInject Malware indicates that xInject is often used by John Smith.
Reverse Relationships			
campaign, intrusion-set	attributed-to	threat-actor	See forward relationship for definition.
indicator	indicates	threat-actor	See forward relationship for definition.

~~2.10.3~~ Examples

```
{
  "type": "threat-actor",
  "id": "threat-actor--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:48.000Z",
  "modified": "2016-04-06T20:03:48.000Z",
  "labels": [ "crime-syndicate"],
  "name": "Evil Org",
  "description": "The Evil Org threat actor group",
  "aliases": ["Syndicate 1", "Evil Syndicate 99"],
  "roles": "director",
  "goals": ["Steal bank money", "Steal credit cards"],
  "sophistication": "advanced",
  "resource_level": "team",
  "primary_motivation": "organizational-gain"
}
```

2.11 Tool

Type Name: tool

Tools are legitimate software that can be used by threat actors to perform attacks. Knowing how and when threat actors use such tools can be important for understanding how campaigns are executed. Unlike malware, these tools or software packages are often found on a system and have legitimate purposes for power users, system administrators, network administrators, or even normal users. Remote access tools (e.g., RDP) and network scanning tools (e.g., Nmap) are examples of Tools that may be used by a Threat Actor during an attack.

The Tool SDO characterizes the properties of these software tools and can be used as a basis for making an assertion about how a Threat Actor uses them during an attack. It contains properties to name and describe the tool, a list of Kill Chain Phases the tool can be used to carry out, and the version of the tool.

This SDO **MUST NOT** be used to characterize malware. Further, Tool **MUST NOT** be used to characteri~~z~~se tools used as part of a course of action in response to an attack. Tools used during response activities can be included directly as part of a Course of Action SDO.

2.11.1 Properties

Common Properties		
type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings		
Tool Specific Properties		
name, description, kill_chain_phases, tool_version		
Property Name	**Type**	**Description**
type (required)	`string`	The value of this property **MUST** be `tool`.
labels (required)	`list` of type `open-vocab`	The kind(s) of tool(s) being described. This is an open vocabulary and values **SHOULD** come from the `tool-label-ov` vocabulary.
name (required)	`string`	The name used to identify the Tool.
description (optional)	`string`	A description that provides more details and context about the Tool, potentially including its purpose and its key characteristics.
kill_chain_phases (optional)	`list` of type `kill-chain-phase`	The list of kill chain phases for which this Tool can be used.
tool_version (optional)	`string`	The version identifier associated with the Tool.

2.11.2 Relationships

These are the relationships explicitly defined between the Tool object and other objects. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Tool object by way of the Relationship object. The reverse relationships (relationships "to" the Tool object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
`created_by_ref`	`identifier` (of type `identity`)
`object_marking_refs`	`identifier` (of type `marking-definition`)

Common Relationships
`duplicate-of`, `derived-from`, `related-to`

Source	Relationship Type	Target	Description
`tool`	`targets`	`identity`, `vulnerability`	This Relationship documents that this Tool is being used to target this Identity or exploit the Vulnerability. For example, a `targets` Relationship linking an exploit Tool to a Vulnerability for CVE-2016-0001 means that the tool exploits that vulnerability. Similarly, a `targets` Relationship linking a DDoS Tool to an Identity representing the energy sector means that Tool is typically used against targets in the energy sector.
Reverse Relationships			
`indicator`	`indicates`	`tool`	See forward relationship for definition
`course-of-action`	`mitigates`	`tool`	See forward relationship for definition

attack-pattern, campaign, intrusion-set, malware, threat-actor	uses	tool	See forward relationship for definition

~~2.11.3~~ Examples

```
{
  "type": "tool",
  "id": "tool--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:03:48.000Z",
  "modified": "2016-04-06T20:03:48.000Z",
  "labels": [ "remote-access"],
  "name": "VNC"
}
```

2.12 Vulnerability

Type Name: vulnerability

A Vulnerability is ""a mistake in software that can be directly used by a hacker to gain access to a system or network"" [CVE]. For example, if a piece of malware exploits CVE-2015-12345, a Malware object could be linked to a Vulnerability object that references CVE-2015-12345.

The Vulnerability SDO is primarily used to link to external definitions of vulnerabilities or to describe 0-day vulnerabilities that do not yet have an external definition. Typically, other SDOs assert relationships to Vulnerability objects when a specific vulnerability is targeted and exploited as part of malicious cyber activity. As such, Vulnerability objects can be used as a linkage to the asset management and compliance process.

2.12.1 Properties

Common Properties		
type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings		
Vulnerability Specific Properties		
name, description		
Property Name	**Type**	**Description**

type (required)	`string`	The value of this property MUST be `vulnerability`.
external_references (optional)	`list` of type `external-reference`	A list of external references which refer to non-STIX information. This property **MAY** be used to provide one or more Vulnerability identifiers, such as a CVE ID [CVE]. When specifying a CVE ID, the **source_name** property of the external reference **MUST** be set to `cve` and the **external_id** property **MUST** be the exact CVE identifier.
name (required)	`string`	A name used to identify the Vulnerability.
description (optional)	`string`	A description that provides more details and context about the Vulnerability, potentially including its purpose and its key characteristics.

2.12.2 Relationships

These are the relationships explicitly defined between the Vulnerability object and other objects. The first section lists the embedded relationships by property name along with their corresponding target. The rest of the table identifies the relationships that can be made from the Vulnerability object by way of the Relationship object. None are defined for the Vulnerability object. The reverse relationships (relationships "to" the Vulnerability object) are included as a convenience. For their definitions, please see the objects for which they represent a "from" relationship.

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

Embedded Relationships	
created_by_ref	`identifier` (of type `identity`)
object_marking_refs	`identifier` (of type `marking-definition`)

Common Relationships			
`duplicate-of`, `derived-from`, `related-to`			
Source	**Relationship Type**	**Target**	**Description**

—	—	—	—
Reverse Relationships			
attack-pattern, campaign, intrusion-set, malware, threat-actor, tool	targets	vulnerability	See forward relationship for definition.
course-of-action	mitigates	vulnerability	See forward relationship for definition.

~~2.12.3~~ Examples

```
{
  "type": "vulnerability",
  "id": "vulnerability--0c7b5b88-8ff7-4a4d-aa9d-feb398cd0061",
  "created": "2016-05-12T08:17:27.000Z",
  "modified": "2016-05-12T08:17:27.000Z",
  "name": "CVE-2016-1234",
  "external_references": [
    {
      "source_name": "cve",
      "external_id": "CVE-2016-1234"
    }
  ]
}
```

3 STIX™ Relationship Objects

STIX Relationship Objects (SROs) represent types of relationships used to describe CTI. The generic Relationship SRO is used to describe many varied types of relationships, while the specific Sighting SRO contains additional properties to represent Sighting relationships.

Property information, relationship information, and examples are provided for each SRO defined below. Property information includes common properties as well as properties that are specific to each SRO. Because SROs cannot be the source or target of other SROs, relationship information is included but only to describe embedded relationships (e.g., **`created_by_ref`**).

3.1 Relationship

Type Name: `relationship`

The Relationship object is used to link together two SDOs in order to describe how they are related to each other. If SDOs are considered "nodes" or "vertices" in the graph, the Relationship Objects (SROs) represent "edges".

STIX defines many relationship types to link together SDOs. These relationships are contained in the "Relationships" table under each SDO definition. Relationship types defined in the specification **SHOULD** be used to ensure consistency. An example of a specification-defined relationship is that an `indicator` `indicates` a `campaign`. That relationship type is listed in the Relationships section of the Indicator SDO definition.

STIX also allows relationships from any SDO to any SDO that have not been defined in this specification. These relationships **MAY** use the `related-to` relationship type or **MAY** use a custom relationship type. As an example, a user might want to link `malware` directly to a `tool`. They can do so using `related-to` to say that the Malware is related to the Tool but not describe how, or they could use `delivered-by` (a custom name they determined) to indicate more detail.

Note that some relationships in STIX may seem like "shortcuts". For example, an Indicator doesn't really detect a Campaign: it detects activity (Attack Patterns, Malware, etc.) that are often used by that campaign. While some analysts might want all of the source data and think that shortcuts are misleading, in many cases it's helpful to provide just the key points (shortcuts) and leave out the low-level details. In other cases, the low-level analysis may not be known or sharable, while the high-level analysis is. For these reasons, relationships that might appear to be "shortcuts" are not excluded from STIX.

3.1.1 Specification-Defined Relationships Summary

This relationship summary table is provided as a convenience. If there is a discrepancy between this table and the relationships defined with each of the SDOs, then the relationships defined with the SDOs **MUST** be viewed as authoritative.

Source	Type	Target	Source	Type	Target
`attack-pattern`	`targets`	`vulnerability`	`intrusion-set`	`attributed-to`	`threat-actor`
`attack-pattern`	`targets`	`identity`	`intrusion-set`	`targets`	`identity`

attack-pattern	uses	malware	intrusion-set	targets	vulnerability
attack-pattern	uses	tool	intrusion-set	uses	attack-pattern
campaign	attributed-to	intrusion-set	intrusion-set	uses	malware
campaign	attributed-to	threat-actor	intrusion-set	uses	tool
campaign	targets	identity	malware	targets	identity
campaign	targets	vulnerability	malware	targets	vulnerability
campaign	uses	attack-pattern	malware	uses	tool
campaign	uses	malware	malware	variant-of	malware
campaign	uses	tool	threat-actor	attributed-to	identity
course-of-action	mitigates	attack-pattern	threat-actor	impersonates	identity
course-of-action	mitigates	malware	threat-actor	targets	identity
course-of-action	mitigates	tool	threat-actor	targets	vulnerability
course-of-action	mitigates	vulnerability	threat-actor	uses	attack-pattern
indicator	indicates	attack-pattern	threat-actor	uses	malware
indicator	indicates	campaign	threat-actor	uses	tool
indicator	indicates	intrusion-set	tool	targets	identity
indicator	indicates	malware	tool	targets	vulnerability
indicator	indicates	threat-actor			
indicator	indicates	tool			

3.1.2 Properties

Common Properties

type, id, created_by_ref, created, modified, revoked, labels, external_references, object_marking_refs, granular_markings		
Relationship Specific Properties		
relationship_type, description, source_ref, target_ref		
Property Name	**Type**	**Description**
type (required)	string	The value of this property **MUST** be relationship.
relationship_type (required)	string	The name used to identify the type of Relationship. This value **SHOULD** be an exact value listed in the relationships for the source and target SDO, but **MAY** be any string. The value of this property **MUST** be in ASCII and is limited to characters a–z (lowercase ASCII), 0–9, and dash (-).
description (optional)	string	A description that provides more details and context about the Relationship, potentially including its purpose and its key characteristics.
source_ref (required)	identifier	The **id** of the source (from) object. The value **MUST** be an ID reference to an SDO (i.e., it cannot point to an SRO, Bundle, or Marking Definition).
target_ref (required)	identifier	The **id** of the target (to) object. The value **MUST** be an ID reference to an SDO (i.e., it cannot point to an SRO, Bundle, or Marking Definition).

3.1.3 Relationships

There are no relationships between the Relationship object and other objects, other than the embedded relationships listed below by property name along with their corresponding target.

Embedded Relationships	
created_by_ref	identifier (of type identity)
object_marking_refs	identifier (of type marking-definition)

3.2 Sighting

Type Name: `sighting`

A Sighting denotes the belief that something in CTI (e.g., an indicator, malware, tool, threat actor, etc.) was seen. Sightings are used to track who and what are being targeted, how attacks are carried out, and to track trends in attack behavior.

The Sighting relationship object is a special type of SRO; it is a relationship that contains extra properties not present on the generic Relationship object. These extra properties are included to represent data specific to sighting relationships (e.g., **`count`**, representing how many times something was seen), but for other purposes a Sighting can be thought of as a Relationship with a name of "sighting-of". Sighting is captured as a relationship because you cannot have a sighting unless you have something that has been sighted. Sighting does not make sense without the relationship to what was sighted.

Sighting relationships relate three aspects of the sighting:

- What was sighted, such as the Indicator, Malware, Campaign, or other SDO (**`sighting_of_ref`**)
- Who sighted it and/or where it was sighted, represented as an Identity (**`where_sighted_refs`**) and
- What was actually seen on systems and networks, represented as Observed Data (**`observed_data_refs`**~~).~~)

What was sighted is required; a sighting does not make sense unless you say what you saw. Who sighted it, where it was sighted, and what was actually seen are optional. In many cases it is not necessary to provide that level of detail in order to provide value.

Sightings are used whenever any SDO has been "seen". In some cases, the object creator wishes to convey very little information about the sighting; the details might be sensitive, but the fact that they saw a malware instance or threat actor could still be very useful. In other cases, providing the details may be helpful or even necessary; saying exactly which of the 1000 IP addresses in an indicator were sighted is helpful when tracking which of those IPs is still malicious.

Sighting is distinct from Observed Data in that Sighting is an intelligence assertion ("I saw this threat actor") while Observed Data is simply information ("I saw this file"). When you combine them by including the linked Observed Data (**`observed_data_refs`**) from a Sighting, you can say "I saw this file, and that makes me think I saw this threat actor". Although **confidence** is currently reserved, notionally confidence would be added to Sighting (the intelligence relationship) but not to Observed Data (the raw information).

3.2.1 Properties

Common Properties
`type`, `id`, `created_by_ref`, `created`, `modified`, `revoked`, `labels`, `external_references`, `object_marking_refs`, `granular_markings`
Sighting Specific Properties

first_seen, last_seen, count, sighting_of_ref, observed_data_refs, where_sighted_refs, summary

Property Name	Type	Description
type (required)	`string`	The value of this property **MUST** be `sighting`.
first_seen (optional)	`timestamp`	The beginning of the time window during which the SDO referenced by the `sighting_of_ref` property was sighted.
last_seen (optional)	`timestamp`	The end of the time window during which the SDO referenced by the `sighting_of_ref` property was sighted.
count (optional)	`integer`	This **MUST** be an integer between 0 and 999,999,999 inclusive and represents the number of times the SDO referenced by the `sighting_of_ref` property was sighted. Observed Data has a similar property called **number_observed**, which refers to the number of times the data was observed. These counts refer to different concepts and are distinct. For example, a single sighting of a DDoS bot might have many millions of observations of the network traffic that it generates. Thus, the Sighting **count** would be 1 (the bot was observed once) but the Observed Data **number_observed** would be much higher. As another example, a sighting with a count of 0 can be used to express that an indicator was not seen at all.
sighting_of_ref (required)	`identifier`	An ID reference to the SDO that was sighted (e.g., Indicator or Malware). For example, if this is a Sighting of an Indicator, that Indicator's ID would be the value of this property.

		This property **MUST** reference only an SDO or a Custom Object.
observed_data_refs (optional)	`list` of type `identifier`	A list of ID references to the Observed Data objects that contain the raw cyber data for this Sighting. For example, a Sighting of an Indicator with an IP address could include the Observed Data for the network connection that the Indicator was used to detect. This property **MUST** reference only Observed Data SDOs.
where_sighted_refs (optional)	`list` of type `identifier`	A list of ID references to the Identity (victim) objects of the entities that saw the sighting. Omitting the **where_sighted_refs** property does not imply that the sighting was seen by the object creator. To indicate that the sighting was seen by the object creator, an Identity representing the object creator should be listed in **where_sighted_refs**. This property **MUST** reference only Identity SDOs.
summary (optional)	`boolean`	The **summary** property indicates whether the Sighting should be considered summary data. Summary data is an aggregation of previous Sightings reports and should not be considered primary source data. Default value is `false`.

3.2.2 Relationships

There are no relationships between the Sighting object and other objects, other than the embedded relationships listed below by property name along with their corresponding target.

Embedded Relationships	
created_by_ref	`identifier` (of type `identity`)

object_marking_refs	identifier (of type marking-definition)
sighting_of_ref	identifier (of type any STIX Object type)
observed_data_refs	list of type identifier (of type observed-data)
where_sighted_refs	list of type identifier (of type identity)

3.2.3 Examples

Sighting of Indicator, without Observed Data

```
{
  "type": "sighting",
  "id": "sighting--ee20065d-2555-424f-ad9e-0f8428623c75",
  "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
  "created": "2016-04-06T20:08:31.000Z",
  "modified": "2016-04-06T20:08:31.000Z",
  "sighting_of_ref": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f"
}
```

Sighting of Indicator, with Observed Data (what exactly was seen) and where it was seen

```
[
  {
    "type": "sighting",
    "id": "sighting--ee20065d-2555-424f-ad9e-0f8428623c75",
    "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
    "created": "2016-04-06T20:08:31.000Z",
    "modified": "2016-04-06T20:08:31.000Z",
    "first_seen": "2015-12-21T19:00:00Z",
    "last_seen": "2015-12-21T19:00:00Z",
    "count": 50,
    "sighting_of_ref": "indicator--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
    "observed_data_refs": ["observed-data--b67d30ff-02ac-498a-92f9-32f845f448cf"],
    "where_sighted_refs": ["identity--b67d30ff-02ac-498a-92f9-32f845f448ff"]
  },
  {
    "type": "observed-data",
    "id": "observed-data--b67d30ff-02ac-498a-92f9-32f845f448cf",
    "created_by_ref": "identity--f431f809-377b-45e0-aa1c-6a4751cae5ff",
    "created": "2016-04-06T19:58:16.000Z",
    "modified": "2016-04-06T19:58:16.000Z",
```

```
    "start": "2015-12-21T19:00:00Z",
    "stop": "2016-04-06T19:58:16Z",
    "count": 50,
    "objects": {
      "0": {
        "type": "file",
        ...
      }
    }
    ...
    }
  }
]
```

4 Conformance

4.1 Object Producers

A "STIX 2.0 Producer" that creates an object from section 2~~2~~ (STIX Domain Objects) or section 3~~3~~ (STIX Relationship Objects) is a "Producer" of that object. Object producers **MUST** conform to all normative requirements in the section for that object.

For example, a "STIX 2.0 Producer" that can produce Indicators is an "Indicator Producer". That producer has to conform to all normative requirements in section 2.5, Indicator.

4.2 Object Consumers

A "STIX 2.0 Consumer" that receives an object from section 2~~2~~ (STIX Domain Objects) or section 3~~3~~ (STIX Relationship Objects) is a "Consumer" of that object. Object consumers **MUST** conform to all normative requirements in the section for that object.

For example, a "STIX 2.0 Consumer" that can receive Campaigns is a "Campaign Consumer". That consumer has to conform to all normative requirements in section 2.2, Campaign.

Appendix A. Glossary

CAPEC - Common Attack Pattern Enumeration and Classification

Consumer - Any entity that receives STIX content

CTI - Cyber Threat Intelligence

Embedded Relationship - A link (an "edge" in a graph) between one STIX Object and another represented as a property on one object containing the ID of another object

Entity - Anything that has a separately identifiable existence (e.g., organization, person, group, etc.)

IEP - FIRST (Forum of Incident Response and Security Teams) Information Exchange Policy

Instance - A single occurrence of a STIX object version

MTI - Mandatory To Implement

MVP - Minimally Viable Product

Object Creator - The entity that created or updated a STIX object (see section 3.3 of *STIX™ Version 2.0. Part 1: STIX Core Concepts*).

Object Representation - An instance of an object version that is serialized as STIX

Producer - Any entity that distributes STIX content, including object creators as well as those passing along existing content

SDO - STIX Domain Object (a "node" in a graph)

SRO - STIX Relationship Object (one mechanism to represent an "edge" in a graph)

STIX - Structured Threat Information Expression

STIX Content - STIX documents, including STIX Objects, STIX Objects grouped as bundles, etc.

STIX Object - A STIX Domain Object (SDO) or STIX Relationship Object (SRO)

STIX Relationship - A link (an "edge" in a graph) between two STIX Objects represented by either an SRO or an embedded relationship

TAXII - An application layer protocol for the communication of cyber threat information

TLP - Traffic Light Protocol

TTP - Tactic, technique, or procedure; behaviors and resources that attackers use to carry out their attacks

~~Appendix A.~~ Appendix B. Acknowledgments

The contributions of the OASIS Cyber Threat Intelligence (CTI) Technical Committee members, enumerated in STIX™ Version 2.0. Part 1: STIX Core Concepts, are gratefully acknowledged.

~~Appendix B.~~Appendix C. Revision History

Revision	Date	Editor	Changes Made
01	2017-01-20	Bret Jordan, John Wunder, Rich Piazza, Ivan Kirillov, Trey Darley	Initial Version
02	2017-04-24	Bret Jordan, John Wunder, Rich Piazza, Ivan Kirillov, Trey Darley	Changes made from first public review