stix2-elevator Documentation

Release 1.0.0

OASIS Open

Oct 08, 2021

Contents:

The stix2-elevator is a software tool for converting STIX 1.x XML to STIX 2.x JSON. Due to the differences between STIX 1.x and STIX 2.x, this conversion is best-effort only, During the conversion, stix2-elevator provides information on the assumptions it needs to make to produce valid STIX 2.x JSON, and what information was not able to be converted.

To convert STIX 2.x JSON back to STIX 1.x XML use the stix2-slider.

For more information about STIX 2, see the website of the OASIS Cyber Threat Intelligence Technical Committee.

CHAPTER 1

Introduction

The stix2-elevator is a python script written to automatically convert STIX 1.x content to STIX 2.x. It is available at https://github.com/oasis-open/cti-stix-elevator/.

The stix2-elevator is a "best-effort" attempt to convert STIX 1.x content to STIX 2.x content. **Caution should be taken if the elevator is to be used in a production environment as warnings concerning the conversion are often generated.** Users should determine which warnings are acceptable and use the –disable option in conjunction with the –error-policy option only to produce results when no other warnings are emitted.

While much of the conversion is straightforward, several assumptions concerning the meaning of the STIX 1.x needed to be made. These are discussed in *Conversion Issues* section.

The elevator produces many messages during the conversion process, that can be reviewed manually to help enhance the automatically produced content, in order to reflect the original content more accurately. A list of these messages can be found in *Warning Messages* section.

CHAPTER 2

Installing

2.1 Requirements

- Python 3.6+
- python-stix and its dependencies

Note: Make sure to use either the latest version of python-stix 1.1.1.x or 1.2.0.x, depending on whether you want to support STIX 1.1.1 or STIX 1.2.

- python-stix2 >= 3.0.0
- stix2-validator >= 3.0.0 and its dependencies
- pycountry >= 20.7.0
- stixmarx >= 1.0.8

2.2 Installation Steps

Install with pip

```
$ pip install stix2-elevator
```

This will install all necessary dependencies, including the latest version of python-stix.

If you need to support older STIX 1.1.1 content, install python-stix 1.1.1.x first

```
$ pip install 'stix<1.2'
$ pip install stix2-elevator
```

You can also install the stix2-elevator from GitHub to get the latest (unstable) version

```
$ pip install git+https://github.com/oasis-open/cti-stix-elevator.git
```

2.3 Installation Steps for ACS Data Marking Support

ACS data markings correspond to the common marking scheme used by the U.S. government (e.g., U, C, S, TS). To elevate STIX 1.x content that contains ACS data markings, it is necessary to install an additional python package called 'stix_edh'.

Install with pip

```
$ pip install stix2-elevator[acs]
```

2.4 Installation Steps for Ignoring Data Markings Not Defined in the STIX Specification

The elevator uses the -m option to declare data marking python classes that support data markings not defined within the STIX specification. See the Command Line Interface section for an example.

However, the elevator must import those class definitions. The suggested way is to create a small python wrapper script that imports the needed package.

```
import <data marking package>
from stix2elevator import elevate

elevate(...)
```

CHAPTER 3

Command Line Interface

The elevator comes with a bundled script which you can use to elevate STIX 1.x content to STIX 2.x content:

```
usage: stix2_elevator [-h]
          [--missing-policy {use-extensions,use-custom-properties,add-to-description,
↪ignore}]
          [--custom-property-prefix CUSTOM_PROPERTY_PREFIX]
          [--infrastructure]
          [--acs]
          [--incidents]
          [--package-created-by-id PACKAGE_CREATED_BY_ID]
          [--default-timestamp DEFAULT_TIMESTAMP]
          [--validator-args VALIDATOR_ARGS]
          [-e ENABLED]
          [-d DISABLED]
          [-s]
          [--message-log-directory MESSAGE_LOG_DIRECTORY]
          [--log-level {DEBUG,INFO,WARN,ERROR,CRITICAL}]
          [-m MARKINGS_ALLOWED]
          [-p {no_policy,strict_policy}]
          [-v {2.0,2.1}]
          file
```

stix2-elevator v4.0.2

positional arguments:

```
file          The input STIX 1.x document to be elevated.
```

optional arguments:

```
-h, --help
              Show this help message and exit

--missing-policy {use-extensions,use-custom-properties,add-to-description,ignore}
              Policy for including STIX 1.x content that cannot be
```

(continues on next page)

(continued from previous page)

```
              represented directly in STIX 2.x. The default is 'add-
              to-description'.

--custom-property-prefix CUSTOM_PROPERTY_PREFIX
              Prefix to use for custom property names when missing
              policy is 'use-custom-properties'. The default is
              'elevator'.

--infrastructure
              Infrastructure will be included in the conversion.
              Default for version 2.1 is true.

--incidents
              Incidents will be included in the conversion.
              Default for version 2.1 is true.

--acs
              Process ACS data markings
              Default is false.

--package-created-by-id PACKAGE_CREATED_BY_ID
              Use provided identifier for "created_by_ref"
              properties.

              Example: --package-created-by-id "identity--1234abcd-1a12-42a3-0ab4-
↪1234abcd5678"

--default-timestamp DEFAULT_TIMESTAMP
              Use provided timestamp for properties that require a
              timestamp.

              Example: --default-timestamp "2016-11-15T13:10:35.053000Z"

--validator-args VALIDATOR_ARGS
              Arguments to pass to stix2-validator.
              See https://stix2-validator.readthedocs.io/en/latest/options.html.

              Example: --validator-args="-v --strict-types -d 212"

-e ENABLED, --enable ENABLED
              A comma-separated list of the stix2-elevator messages
              to enable. If the --disable option is not used, no
              other messages will be shown.

              Example: --enable 250

-d DISABLED, --disable DISABLED
              A comma-separated list of the stix2-elevator messages
              to disable.

              Example: --disable 212,220

-s, --silent
              If this flag is set, all stix2-elevator messages will
              be disabled.

--message-log-directory MESSAGE_LOG_DIRECTORY
```

(continues on next page)

(continued from previous page)

```
              If this flag is set, all stix2-elevator messages will
              be saved to a file. The name of the file will be the
              input file with extension .log in the specified
              directory.

              Note, make sure the directory already exists.

              Example: --message-log-directory "../logs".

--log-level {DEBUG,INFO,WARN,ERROR,CRITICAL}
              The logging output level.

-m MARKINGS_ALLOWED, --markings-allowed MARKINGS_ALLOWED
              Avoid error exit, if these markings types
              (as specified via their python class names) are in the
              content, but not supported by the elevator. Specify as
              a comma-separated list.

              Example: --markings-allowed "ISAMarkingsAssertion,ISAMarkings"

-p {no_policy,strict_policy},
--error-policy {no_policy,strict_policy},
--policy {no_policy,strict_policy}   #deprecated
             The policy to deal with errors. The default is 'no_policy'.

-v {2.0,2.1}, --version {2.0,2.1}
             The version of stix 2 to be produced. The default is 2.1
```

Refer to the *Warning Messages* section for all stix2-elevator messages. Use the associated code number to `--enable` or `--disable` a message. By default, the stix2-elevator displays all messages.

Note: disabling the message does not disable any functionality.

CHAPTER 4

Mappings from STIX 1.x to STIX 2.x

This section outlines the disposition of each property of the top-level objects when converted.

For each STIX 1.x object that was converted the following options are possible:

- **STIX 1.x property mapped directly to a STIX 2.x property.** This property's value is used unaltered in the conversion to 2.x.
- **STIX 1.x property translated into STIX 2.x property.** This property's value must undergo some minor processing to determine the corresponding content for 2.x.
- **STIX 1.x property mapped using STIX 2.x relationships.** This property is used to construct a 2.x relationship object. The "reverse" notation indicates the the STIX 1.x property is found on target object.
- **STIX 1.x property handled based on the "missing policy" option.** This property has no corresponding property in STIX 2.x, but its value can be (optionally) included using the extension mechanism, custom properties or in the description property of the 2.x object as text, depending upon the **–missing-policy** option.
- **STIX 1.x property not mapped.** This property will not be included in the converted 2.x object.

All examples were generated using the missing policy of **add-to-description**.

4.1 Top Level Object Mappings

This table describes the mapping between STIX 1.x and STIX 2.x top-level objects. Notice that certain object types in STIX 1.x that were not top-level objects are in STIX 2.x (e.g., Malware). In STIX 2.1, cyber observable objects are also top-level objects - but their mapping can be found in the *Mappings from CybOX 2.x to STIX 2.x* section

STIX 1.x object	STIX 2.x object
Campaign	campaign
Course_Of_Action	course-of-action
et:Vulnerability	vulnerability
et:Weakness	*not converted*
et:Configuration	*not converted*
Incident	incident *in 2.1*
Indicator	indicator
Information_Source/ CIQIdentity3_0Instance/ Address	location *in 2.1*
Report	report
Observable	observed-data
Package	bundle
Threat Actor	threat-actor
ttp:Attack_Pattern	attack-pattern
ttp:Infrastructure	infrastructure
ttp:Malware	malware
ttp:Persona	*not converted*
ttp:Tool	tool
ttp:Victim_Targeting	identity

4.2 Common Properties

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
Description	description
timestamp	modified
Title	name

In STIX 1.x only one timestamp is recorded, whereas in STIX 2.x, there are two properties: created and modified. The created timestamp is not stored in objects in STIX 1.x. The timestamp property in STIX 1.x holds the modified timestamp.

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
id	id
Handling	object_markings_refs, granular_markings
Information_Source	created_by_ref, external_references
Confidence	confidence

In STIX 1.x, an id contained a "namespace". This was deemed unnecessary in STIX 2.x, therefore they contain no origin information.

- Handling

  Data Markings, called Handling in STIX 1.x, have been completely redesigned for STIX 2.x. STIX 1.x used *xpath*, which was a reasonable choice given its reliance on XML for implementation. However, the use of xpath was very difficult to implement, and was more expressive than was deemed necessary.

STIX 2.x introduces two new concepts, object markings and granular markings, which simplify the marking of data. Object markings apply to a whole object, whereas granular markings are specific to particular properties of an object. The selection of which properties are to be marked is expressed in a serialization-neutral way. The scope of marking definitions is at the object level. There is no marking that can apply to a whole bundle, or report.

- Information_Source

  In STIX 1.x there were several related concepts that were used to identify the sources of information and various parties of interest. Parties of interest are creators of content, victim targets, and other responsible parties. Sources of information could be an individual, organization or some software application. Additionally, it was possible to make references to source material external to STIX, e.g., a citation, URL, or an ID in an external system or repository.

  In STIX 2.x, we have retained the concept of an `IdentityType` object, but do not rely on the OASIS CIQ standard model as STIX 1.x did. The `Identity` object type in STIX 2.x contains a very streamlined set of properties: `identity_class` to specify if it is an individual or organization, `sectors` to indicate the industry sector that the identity belongs to, and a free text property, `contact_information` to specify such information. Other OASIS CIQ standard model propeties are not mapped in the conversion.

  The `InformationSourceType` object was used in STIX 1.x to associate an object with its creator's identity. In STIX 2.x, the common property `created_by_ref` is used, and it must contain the identifier of an `Identity` object.

  The `InformationSourceType` object was also used in STIX 1.x to specify external information. Other properties like `capec_id` of `AttackPatternType`, or `cve_id` of `VulnerabilityType` were also used for external information, holding the ids of items in repositories or systems external to STIX. In STIX 2.x, the data type `external-reference` is used for all external information.

  The `InformationSourceType` object was also used in STIX 1.x to specify location information. The `location` object will be used when converting to STIX 2.1.

- Type

  In STIX 2.x, the type of an object is defined to be a specific literal, and is recorded in the `type` property. The type of an object in STIX 1.x was either implicitly defined by its element name or explicitly using xsi:type.

- Kill Chains

  In STIX 1.x, kill chains, with their phases, were defined using the `KillChainType`, which is found in the `Kill_Chains` property of a `TTP`. These kill chains phases were refered to in the `TTP` and `Indicator` `Kill_Chain_Phases` properties. In STIX 2.x, kill chains and their phases are not explicitly defined, but are referenced using their common names.

  If the Lockheed Martin Cyber Kill Chain™ is used the `kill_chain_name` property must be `lockheed-martin-cyber-kill-chain`, according to the specification.

STIX 1.x Properties Mapped Using STIX 2.x Relationships

none

STIX 1.x Properties Handled Based by the "missing policy"

- `Short_Description`
- `Confidence` *in STIX 2.0*

  The confidence concept is available only STIX 2.1.

In the examples, the missing policy, if used, is `add-to-description`, or `use-extensions` for some 2.1 examples.

STIX 1.x Properties Not Mapped

- `idref`

  Relationships in STIX 2.x make use of id references to indicate the source and target of the relationship. STIX 2.x objects additionally use `id` references for any property whose suffix is `ref` or `refs`. The facility available in STIX 1.x to specify related objects by embedding them in other objects is not available in STIX 2.x.

- `Related_Packages`

  STIX 1.x packages correspond to STIX 2.x bundles. However, bundles cannot refer to other bundles, so there is no way to express this property in STIX 2.x.

- `Version`

  Individual STIX objects do not have their own STIX version in STIX 2.0. A bundle has the property `spec_version`, which applies to all objects that are contained in the bundle. In STIX 2.1, objects do have the property `spec_version`. In all cases, the version information is not transfered from the STIX 1.x object, but depends upon the –version option when invoking the elevator.

  In the examples below, the `spec_version` property is omitted, but for STIX 2.1 it is often required.

4.2.1 Versioning

STIX 1.x supported the versioning of objects, but it was a feature that was rarely used. STIX 2.x support of versioning is based on two common properties: `modified` and `revoked`. However, the elevator does not support converting STIX 1.x versioned objects, in the unlikely inclusion of such objects.

All converted objects will be assumed to be the one and only version of an object. If more than one object is found with the same id, it will *not* be flagged as an error.

4.3 Relationships

All STIX 1.x relationships were defined explicitly in the specification and they are all embedded as properties of the object. In STIX 2.x, relationships are top-level objects so they exist independently from their source and target objects. Additionally, although the STIX 2.x specification suggests certain relationships between object types, a relationship between any two objects is allowed.

Relationships in STIX 1.x could be specified either using the `idref` property, or by embedding the object within the relationship itself. In the former case, the STIX 2.x object should use the original object's `id` as the `source_ref` property, and the `idref` as the `target_ref` property. In the latter case, the embedded object must first be converted to a top-level STIX 2.x object. Of course, the embedded object's `id` might not present. In that case, an new id must be created.

An Example

STIX 1.x in XML

```xml
<stix:Campaign id="example:Campaign-e5268b6e-4931-42f1-b379-87f48eb41b1e"
               timestamp="2014-08-08T15:50:10.983728+00:00"
               xsi:type='campaign:CampaignType' version="1.2">
    <campaign:Attribution>
        <campaign:Attributed_Threat_Actor>
```

(continues on next page)

(continued from previous page)

```
            <stixCommon:Threat_Actor idref="example:threatactor-56f3f0db-b5d5-431c-
↪ae56-c18f02caf500"/>
        </campaign:Attributed_Threat_Actor>
    </campaign:Attribution>
</stix:Campaign>
```

STIX 2.x in JSON

```
{
        "created": "2014-08-08T15:50:10.983Z",
        "id": "relationship--3dcf59c3-30e3-4aa5-9c05-2cbffcee5922",
        "modified": "2014-08-08T15:50:10.983Z",
        "relationship_type": "attributed-to",
        "source_ref": "campaign--e5268b6e-4931-42f1-b379-87f48eb41b1e",
        "target_ref": "threat-actor--56f3f0db-b5d5-431c-ae56-c18f02caf500",
        "type": "relationship"
}

{
        "id": "campaign--e5268b6e-4931-42f1-b379-87f48eb41b1e"

}

{
        "id": "threat-actor--56f3f0db-b5d5-431c-ae56-c18f02caf500"

}
```

4.4 Attack Pattern

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

none

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
capec_id	external_references
ttp:Kill_Chain_Phases	kill_chain_phases

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
ttp:Victim_Targeting	targets
ttp:Exploit_Targets	targets (vulnerability, only)
ttp:Related_TTPs	uses (malware, tool), related-to (when not used for versioning)

STIX 1.x Properties Handled Based on the "missing policy"

- ttp:Intended_Effect

STIX 1.x Properties Not Mapped

- ttp:Kill_Chains

An Example

STIX 1.x in XML

```
<stix:TTP id="example:ttp-8ac90ff3-ecf8-4835-95b8-6aea6a623df5" xsi:type='ttp:TTPType
↪'>
   <ttp:Title>Phishing</ttp:Title>
   <ttp:Behavior>
       <ttp:Attack_Patterns>
           <ttp:Attack_Pattern capec_id="CAPEC-98">
               <ttp:Description>Phishing</ttp:Description>
           </ttp:Attack_Pattern>
       </ttp:Attack_Patterns>
   </ttp:Behavior>
   <ttp:Information_Source>
       <stixCommon:Identity idref="example:identity-f690c992-8e7d-4b9a-9303-
↪3312616c0220"/>
   </ttp:Information_Source>
</stix:TTP>
```

STIX 2.x in JSON

```
{
   "created": "2017-01-27T13:49:54.326Z",
   "created_by_ref": "identity--f690c992-8e7d-4b9a-9303-3312616c0220"
   "description": "Phishing",
   "external_references": [
       {
           "external_id": "CAPEC-98",
           "source_name": "capec"
       }
   ],
   "id": "attack-pattern--8ac90ff3-ecf8-4835-95b8-6aea6a623df5",
   "modified": "2017-01-27T13:49:54.326Z",
   "name": "Phishing",
   "type": "attack-pattern"
}
```

4.5 Campaigns

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
Names	aliases

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
Intended_Effect	objective

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
Related_TTPs	uses
Related_Campaign	indicates (reverse)
Attribution	attributed-to
Associated_Campaigns	related-to (when not used for versioning)

STIX 1.x Properties Handled Based on the "missing policy"

- Status

STIX 1.x Properties Not Mapped

- Activity
- Related_Incidents

An Example

STIX 1.x in XML

```
<stix:Campaign id="example:Campaign-e5268b6e-4931-42f1-b379-87f48eb41b1e"
               timestamp="2014-08-08T15:50:10.983"
               xsi:type='campaign:CampaignType' version="1.2">
    <campaign:Title>Operation Bran Flakes</campaign:Title>
    <campaign:Description>A concerted effort to insert false information into the BPP
↪'s web pages</campaign:Description>
    <campaign:Names>
        <campaign:Name>OBF</campaign:Name>
    </campaign:Names>
    <campaign:Intended_Effect>Hack www.bpp.bn</campaign:Intended_Effect>
    <campaign:Related_TTPs>
        <campaign:Related_TTP>
            <stixCommon:TTP id="example:ttp-2d1c6ab3-5e4e-48ac-a32b-f0c01c2836a8"
                            timestamp="2014-08-08T15:50:10.983464+00:00"
                            xsi:type='ttp:TTPType' version="1.2">
                 <ttp:Victim_Targeting>
                     <ttp:identity id="example:identity-ddfe7140-2ba4-48e4-b19a-
↪df069432103b">
                        <stixCommon:name>Branistan Peoples Party</stixCommon:name>
                    </ttp:identity>
                 </ttp:Victim_Targeting>
             </stixCommon:TTP>
         </campaign:Related_TTP>
    </campaign:Related_TTPs>
    <campaign:Attribution>
         <campaign:Attributed_Threat_Actor>
             <stixCommon:Threat_Actor idref="example:threatactor-56f3f0db-b5d5-431c-
↪ae56-c18f02caf500"/>
         </campaign:Attributed_Threat_Actor>
    </campaign:Attribution>
    <campaign:Information_Source>
        <stixCommon:Identity id="example:identity-f690c992-8e7d-4b9a-9303-3312616c0220
↪">
        <stixCommon:name>The MITRE Corporation - DHS Support Team</stixCommon:name>
        <stixCommon:Role xsi:type="stixVocabs:InformationSourceRoleVocab-1.0">Initial␣
↪Author</stixCommon:Role>
   </campaign:Information_Source>
</stix:Campaign>
```

STIX 2.x in JSON

```
{
    "type": "identity",
    "id": "identity--f690c992-8e7d-4b9a-9303-3312616c0220",
    "created": "2016-08-08T15:50:10.983Z",
    "modified": "2016-08-08T15:50:10.983Z",
    "name": "The MITRE Corporation - DHS Support Team",
    "identity_class": "organization"
}

{
    "type": "identity",
    "id": "identity--ddfe7140-2ba4-48e4-b19a-df069432103b",
    "created_by_ref": "identity--f690c992-8e7d-4b9a-9303-3312616c0220",
    "created": "2016-08-08T15:50:10.983Z",
    "modified": "2016-08-08T15:50:10.983Z",
    "name": "Branistan Peoples Party",
    "identity_class": "organization"
}

{
    "type": "campaign",
    "id": "campaign--e5268b6e-4931-42f1-b379-87f48eb41b1e",
    "created_by_ref": "identity--f690c992-8e7d-4b9a-9303-3312616c0220",
    "created": "2016-08-08T15:50:10.983Z",
    "modified": "2016-08-08T15:50:10.983Z",
    "name": "Operation Bran Flakes",
    "description": "A concerted effort to insert false information into the BPP's web pages",
    "aliases": ["OBF"],
    "first_seen": "2016-01-08T12:50:40.123Z",
    "objective": "Hack www.bpp.bn"
}
```

See *Threat Actor* for the Threat Actor object.

4.6 Course of Action

In STIX 2.x the `course-of-action` object is defined as a stub. This means that in STIX 2.x this object type is pretty “bare-bones”, not containing most of the properties that were found in STIX 1.x. The property `action` is reserved, but not defined in STIX 2.x.

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Type`	`labels`

STIX 1.x Properties Translated to STIX 2.x Properties

none

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
`Related_COAs`	`related-to` (when not used for versioning)

STIX 1.x Properties Handled Based on the "missing policy"

- Stage
- Objective
- Impact
- Cost
- Efficacy
- Parameter_Observables

STIX 1.x Properties Not Mapped

- Structured_COA

An Example

STIX 1.x in XML

```
<stix:Course_Of_Action id="example:coa-495c9b28-b5d8-11e3-b7bb-000c29789db9" xsi:type=
↪'coa:CourseOfActionType' version="1.2">
    <coa:Title>Block traffic to PIVY C2 Server (10.10.10.10)</coa:Title>
    <coa:Stage xsi:type="stixVocabs:COAStageVocab-1.0">Response</coa:Stage>
    <coa:Type xsi:type="stixVocabs:CourseOfActionTypeVocab-1.0">Perimeter Blocking</
↪coa:Type>
    <coa:Objective>
        <coa:Description>Block communication between the PIVY agents and the C2 Server
↪</coa:Description>
        <coa:Applicability_Confidence>
            <stixCommon:Value xsi:type="stixVocabs:HighMediumLowVocab-1.0">High</
↪stixCommon:Value>
        </coa:Applicability_Confidence>
    </coa:Objective>
    <coa:Parameter_Observables cybox_major_version="2" cybox_minor_version="1" cybox_
↪update_version="0">
        <cybox:Observable id="example:Observable-356e3258-0979-48f6-9bcf-6823eecf9a7d
↪">
            <cybox:Object id="example:Address-df3c710c-f05c-4edb-a753-de4862048950">
                <cybox:Properties xsi:type="AddressObj:AddressObjectType" category=
↪"ipv4-addr">
                    <AddressObj:Address_Value>10.10.10.10</AddressObj:Address_Value>
                </cybox:Properties>
            </cybox:Object>
        </cybox:Observable>
    </coa:Parameter_Observables>
    <coa:Impact>
        <stixCommon:Value xsi:type="stixVocabs:HighMediumLowVocab-1.0">Low</
↪stixCommon:Value>
        <stixCommon:Description>This IP address is not used for legitimate hosting so␣
↪there should be no operational impact.</stixCommon:Description>
    </coa:Impact>
    <coa:Cost>
        <stixCommon:Value xsi:type="stixVocabs:HighMediumLowVocab-1.0">Low</
↪stixCommon:Value>
    </coa:Cost>
    <coa:Efficacy>
        <stixCommon:Value xsi:type="stixVocabs:HighMediumLowVocab-1.0">High</
↪stixCommon:Value>
```

(continues on next page)

(continued from previous page)

```
    </coa:Efficacy>
</stix:Course_Of_Action>
```

STIX 2.x in JSON

```
{
    "id": "bundle--495c4c04-b5d8-11e3-b7bb-000c29789db9",
    "objects": [
        {
            "created": "2017-01-27T13:49:41.298Z",
            "description": "\n\nSTAGE:\n\tResponse\n\n
                                OBJECTIVE: Block communication between the PIVY agents and the C2 Server\n\n
                                CONFIDENCE: High\n\n
                                IMPACT:Low, This IP address is not used for legitimate hosting so there should be no operational impact.\n\n
                                COST:Low\n\n
                                EFFICACY:High",
            "id": "course-of-action--495c9b28-b5d8-11e3-b7bb-000c29789db9",
            "labels": [
                "perimeter-blocking"
            ],
            "modified": "2017-01-27T13:49:41.298Z",
            "name": "Block traffic to PIVY C2 Server (10.10.10.10)",
            "type": "course-of-action"
        }
    ],
    "spec_version": "2.0",
    "type": "bundle"
}
```

Notice that the `spec_version` property only appears on the bundle in STIX 2.0, but in STIX 2.1, it is *not* a property of the bundle. It may (optionally) appear on each object. The elevator will always provides the `spec_version` property for all 2.1 SDOs and SROs, but not on SCOs.

4.7 Incident

In STIX 2.1 the `Incident` object is defined as a stub. This means that in STIX 2.x this object type is pretty "bare-bones", not containing most of the properties that were found in STIX 1.x.

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

none

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Categories`	`labels`
`External_ID`	`external_references`

STIX 1.x Properties Mapped Using STIX 2.x Relationships

- `Related_Indicators`
- `Related_Observables`

- Leveraged_TTPs
- Attributed_Threat_Actors
- COA_Requested
- COA_Taken

STIX 1.x Properties Handled Based on the “missing policy”

- Reporter
- Responder
- Coordinator
- Victims
- Status
- Contact
- Intended_Effect

STIX 1.x Properties Not Mapped

- Affected_Assets
- Impact_Assessment
- History
- URL
- Time

An Example

STIX 1.x in XML

```
<stix:Incidents>
    <stix:Incident id="example:incident-1b75ee8f-44d6-819a-d729-09ab52c91fdb"␣
↪xsi:type='incident:IncidentType' timestamp="2014-05-08T09:00:00.000000Z">
        <incident:Title>Detected Poison Ivy beaconing through perimeter firewalls</
↪incident:Title>
        <incident:Status>New</incident:Status>
        <incident:Contact>
            <stixCommon:Identity>
                <stixCommon:Name>Fred</stixCommon:Name>
            </stixCommon:Identity>
        </incident:Contact>
        <incident:Contact>
            <stixCommon:Identity>
                <stixCommon:Name>Barney</stixCommon:Name>
            </stixCommon:Identity>
        </incident:Contact>
        <incident:Leveraged_TTPs>
            <incident:Leveraged_TTP>
                <stixCommon:Relationship>Uses Malware</stixCommon:Relationship>
                <stixCommon:TTP idref="example:ttp-e610a4f1-9676-4ab3-bcc6-
↪b2768d58281b"/>
            </incident:Leveraged_TTP>
        </incident:Leveraged_TTPs>
    </stix:Incident>
</stix:Incidents>
```

STIX 2.1 in JSON

```
{
    "id": "bundle--65184e82-b693-41e3-bfd7-0800271e87d2",
    "objects": [
        {
            "created": "2014-05-08T09:00:00.000Z",
            "id": "identity--8e5febda-ffd0-4ade-8afe-9a7e64894510",
            "modified": "2014-05-08T09:00:00.000Z",
            "name": "Fred",
            "spec_version": "2.1",
            "type": "identity"
        },
        {
            "created": "2014-05-08T09:00:00.000Z",
            "id": "identity--b2557302-99e3-496a-825f-8e8c5501bec8",
            "modified": "2014-05-08T09:00:00.000Z",
            "name": "Barney",
            "spec_version": "2.1",
            "type": "identity"
        },
        {
            "created": "2014-05-08T09:00:00.000Z",
            "extensions": {
                "extension-definition--7a8eaf47-9b0f-487d-b280-1e6cc4cccee9": {
                    "contacts": [
                        "identity--8e5febda-ffd0-4ade-8afe-9a7e64894510",
                        "identity--b2557302-99e3-496a-825f-8e8c5501bec8"
                    ],
                    "extension_type": "property-extension",
                    "status": "New"
                }
            },
            "id": "incident--1b75ee8f-44d6-819a-d729-09ab52c91fdb",
            "modified": "2014-05-08T09:00:00.000Z",
            "name": "Detected Poison Ivy beaconing through perimeter firewalls",
            "spec_version": "2.1",
            "type": "incident"
        },
        {
            "created": "2014-05-08T09:00:00.000Z",
            "description": "Uses Malware",
            "id": "relationship--d695b661-62ff-4685-bf88-a449770969ed",
            "modified": "2014-05-08T09:00:00.000Z",
            "relationship_type": "related-to",
            "source_ref": "incident--1b75ee8f-44d6-819a-d729-09ab52c91fdb",
            "spec_version": "2.1",
            "target_ref": "malware--6516102d-b693-41e3-bfd7-0800271e87d2",
            "type": "relationship"
        }
    ],
    "type": "bundle"
}
```

4.8 Indicator

STIX 1.x Composite Indicator Expressions and CybOX 2.x Composite Observable Expressions allow a level of flexibility not present in STIX 2.x patterns. These composite expressions can frequently have ambiguous interpretations, so STIX 2.x Indicators created by the stix2-elevator from STIX 1.x Indicators containing composite expressions should be inspected to ensure the STIX 2.x Indicator has the intended meaning.

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Valid_Time_Position`	`valid_from`, `valid_until`
`Type`	`labels` in 2.0, `indicator_type` in 2.1

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Alternative_ID`	`external_references`
`Kill_Chain_Phases`	`kill_chain_phases`
`Observable` `Composite_Indicator_Expression`	`pattern`
`Test_Mechanisms`	`pattern`
`Producer`	`created_by_ref`

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
`Indicated_TTP`	`detects`
`Suggested_COAs`	`related-to`
`Related_Indicators`	`related-to` (when not used for versioning)
`Related_Campaigns`	`indicates`

STIX 1.x Properties Handled Based on the "missing policy"

- `Likely_Impact`

STIX 1.x Properties Not Mapped

- `negate`

An Example

STIX 1.x in XML

```
<stix:Indicator id="example:Indicator-d81f86b9-975b-bc0b-775e-810c5ad45a4f"
                xsi:type='indicator:IndicatorType'>
    <indicator:Title>Malicious site hosting downloader</indicator:Title>
    <indicator:Type xsi:type="stixVocabs:IndicatorTypeVocab-1.0">URL Watchlist</
↪indicator:Type>
    <indicator:Observable id="example:Observable-ee59c28e-d922-480e-9b7b-a79502696505
↪">
        <cybox:Object id="example:URI-b13ae3fc-80af-49c2-9de9-f713abc070ba">
            <cybox:Properties xsi:type="URIObj:URIObjectType" type="URL">
                <URIObj:Value condition="Equals">http://x4z9arb.cn/4712</URIObj:Value>
            </cybox:Properties>
        </cybox:Object>
    </indicator:Observable>
</stix:Indicator>
```

STIX 2.1 in JSON

```
{
   "created": "2017-01-27T13:49:53.935Z",
   "id": "indicator--d81f86b9-975b-bc0b-775e-810c5ad45a4f",
   "indicator_types": [
       "url-watchlist"
   ],
   "modified": "2017-01-27T13:49:53.935Z",
   "name": "Malicious site hosting downloader",
   "pattern": "[url:value = 'http://x4z9arb.cn/4712']",
   "pattern_type": "stix",
   "spec_version": "2.1",
   "type": "indicator",
   "valid_from": "2017-01-27T13:49:53.935382Z"
}
```

`indicator_types` would be `labels` and `pattern_type` is not used in 2.0

Sightings

In STIX 1.x sightings were a property of `IndicatorType`. In STIX 2.x, sightings are a top-level STIX *relationship* object. Because they represent the relationship (match) of an indicator pattern to observed data (or other object), they are more naturally represented as a STIX 2.x relationship.

For example, suppose the above indicator pattern was matched against an actual cyber observable ("observed-data–b67d30ff-02ac-498a-92f9-32f845f448cf"), because a victim (whose identity is represented by "identity–b67d30ff-02ac-498a-92f9-32f845f448ff") observed that URL.

The STIX 2.x sighting would be:

```
{
    "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--d81f86b9-975b-bc0b-775e-810c5ad45a4f",
    "observed_data_refs": ["observed-data--b67d30ff-02ac-498a-92f9-32f845f448cf"],
    "where_sighted_refs": ["identity--b67d30ff-02ac-498a-92f9-32f845f448ff"]
}
```

4.9 Infrastructure

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Type`	`labels` in 2.0, `infrastructure_types` in 2.1

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`ttp:Kill_Chain_Phases`	`kill_chain_phases`

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
`Observable_Characterizations`	`consists_of`
`ttp:Exploit_Targets`	`has` (vulnerability, only)
`ttp:Related_TTPs`	`delivers` (malware), `related-to` (when not used for versioning)

STIX 1.x Properties Handled Based on the "missing policy"

none

STIX 1.x Properties Not Mapped

none

An Example

STIX 1.x in XML

```
<stix:TTP xsi:type="ttp:TTPType" id="example:ttp-dd955e08-16d0-4f08-5064-50d9e7a3104d
↪" timestamp="2014-05-08T09:00:00.000000Z">
        <ttp:Title>Malware C2 Channel</ttp:Title>
        <ttp:Resources>
            <ttp:Infrastructure>
                <ttp:Type>Malware C2</ttp:Type>
                <ttp:Observable_Characterization cybox_major_version="2" cybox_minor_
↪version="1">
                    <cybox:Observable id="example:observable-c8c32b6e-2ea8-41c4-6446-
↪7f5218072f27">
                        <cybox:Object id="example:object-d7fcce87-0e98-4537-81bf-
↪1e7ca9ad3734">
                            <cybox:Properties xsi:type="FileObj:FileObjectType">
                                <FileObj:File_Name>iprip32.dll</FileObj:File_Name>
                            </cybox:Properties>
                        </cybox:Object>
                    </cybox:Observable>
                </ttp:Observable_Characterization>
            </ttp:Infrastructure>
        </ttp:Resources>
    </stix:TTP>
</stix:TTPs>
```

STIX 2.1 in JSON

```
{
    "id": "bundle--cc0ca596-70e6-4dac-9bef-603166d17db8",
    "objects": [
        {
            "id": "file--bccadc39-2701-5c0b-8abd-fb2efd61c6be",
            "name": "iprip32.dll",
            "type": "file"
        },
        {
            "created": "2014-05-08T09:00:00.000Z",
            "first_seen": "2014-05-08T09:00:00.000Z",
            "id": "infrastructure--63d4313e-437e-4ed1-a8b4-aa04d95f1c18",
            "infrastructure_types": [
                "malware-c2"
```

(continues on next page)

(continued from previous page)

```
            ],
            "modified": "2014-05-08T09:00:00.000Z",
            "name": "Malware C2 Channel",
            "spec_version": "2.1",
            "type": "infrastructure"
        },
        {
            "created": "2014-05-08T09:00:00.000Z",
            "id": "relationship--3b86f807-ebdf-47db-88ac-5d13b2b8028b",
            "modified": "2014-05-08T09:00:00.000Z",
            "relationship_type": "consists-of",
            "source_ref": "infrastructure--63d4313e-437e-4ed1-a8b4-aa04d95f1c18",
            "spec_version": "2.1",
            "target_ref": "file--bccadc39-2701-5c0b-8abd-fb2efd61c6be",
            "type": "relationship"
        }
    ],
    "type": "bundle"
}
```

4.10 Location

In STIX 2.1 the `location` object corresponds to any `Information_Source` Address objects in STIX 1.x. `Information_Source` objects with `Address` information can appear in most top-level STIX 1.x objects. However, you cannot store location information as a property in STIX 2.1, because `location` is a top-level object. To do the conversion, it is necessary to create a new STIX 2.1 `location` object, transfering the STIX 1.x address information into it, and introducing a STIX 2.x `relationship` object between that original object and the new `location` object.

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x relationship type
`Administrative_Area`	`administrative_area`
`Country`	`country`

STIX 1.x Properties Translated to STIX 2.x Properties

none

STIX 1.x Properties Mapped Using STIX 2.x Relationships

none

STIX 1.x Properties Handled Based on the "missing policy"

- `free_text_address`

STIX 1.x Properties Not Mapped

none

An Example

STIX 1.x in XML

```
<ta:Identity id="example:Identity-733c5838-34d9-4fbf-949c-62aba761184c" xsi:type=
↪'stix-ciqidentity:CIQIdentity3.0InstanceType'>
    <ExtSch:Specification xmlns:ExtSch="http://stix.mitre.org/extensions/Identity
↪#CIQIdentity3.0-1">
        <xpil:PartyName xmlns:xpil="urn:oasis:names:tc:ciq:xpil:3">
            <xnl:OrganisationName xmlns:xnl="urn:oasis:names:tc:ciq:xnl:3" xnl:Type=
↪"CommonUse">
                <xnl:NameElement>Disco Tean</xnl:NameElement>
            </xnl:OrganisationName>
            <xnl:OrganisationName xmlns:xnl="urn:oasis:names:tc:ciq:xnl:3" xnl:Type=
↪"UnofficialName">
                <xnl:NameElement>Equipo del Discoteca</xnl:NameElement>
            </xnl:OrganisationName>
        </xpil:PartyName>
        <xpil:Addresses xmlns:xpil="urn:oasis:names:tc:ciq:xpil:3">
            <xpil:Address>
                <xal:Country xmlns:xal="urn:oasis:names:tc:ciq:xal:3">
                    <xal:NameElement>United States</xal:NameElement>
                </xal:Country>
                <xal:AdministrativeArea xmlns:xal="urn:oasis:names:tc:ciq:xal:3">
                    <xal:NameElement>California</xal:NameElement>
                </xal:AdministrativeArea>
            </xpil:Address>
        </xpil:Addresses>
    </ExtSch:Specification>
</ta:Identity>
```

STIX 2.1 in JSON

```
{
    "id": "bundle--ccd00c4a-1bdb-46ae-9898-ecaca13f1f12",
    "objects": [
        {
          "administrative_area": "California",
          "country": "US",
          "created": "2014-11-19T23:39:03.893Z",
          "id": "location--c1445467-fd92-4532-9161-1c3024ab6467",
          "modified": "2014-11-19T23:39:03.893Z",
          "spec_version": "2.1",
          "type": "location"
        },
        {
          "created": "2014-11-19T23:39:03.893Z",
          "id": "relationship--b1d9c097-a0ac-46e8-997b-291ea3b976f5",
          "modified": "2014-11-19T23:39:03.893Z",
          "relationship_type": "located-at",
          "source_ref": "identity--733c5838-34d9-4fbf-949c-62aba761184c",
          "spec_version": "2.1",
          "target_ref": "location--c1445467-fd92-4532-9161-1c3024ab6467",
          "type": "relationship"
        },
        {
          "created": "2014-11-19T23:39:03.893Z",
          "id": "identity--733c5838-34d9-4fbf-949c-62aba761184c",
          "identity_class": "organization",
          "modified": "2014-11-19T23:39:03.893Z",
          "name": "Disco Tean",
```

(continues on next page)

(continued from previous page)

```
            "spec_version": "2.1",
            "type": "identity"
        }
    ],
    "type": "bundle"
}
```

4.11 Malware

The Malware object in STIX 1.x is a stub, which depends up MAEC content for further properties. The elevator does not support the conversion of MAEC content. The main properties of malware in STIX 2.0 are not much different than the defined ones in 1.x. STIX 2.1 included more properties, and additionally the object type `malware-analysis`, therefore conversion of MAEC content could be supported in a future release of the elevator.

Malware is not a top-level object in STIX 1.x, but a property of a `TTP`.

The `name` property of the STIX 1.x Malware object is the preferred property to use to populated the `name` property in the STIX 2.x object, although if missing, the `title` property can be used.

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Type`	`labels` in 2.0, `malware_types` in 2.1

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`ttp:Kill_Chain_Phases`	`kill_chain_phases`

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
`ttp:Related_TTPs`	`variant-of` (malware), `related-to` (when not used for versioning), uses (tool)
`ttp:Exploit_Targets`	`targets` (vulnerability, only)
`ttp:Victim_Targeting`	`targets`

STIX 1.x Properties Handled Based on the “missing policy”

- `ttp:Intended_Effect`

STIX 1.x Properties Not Mapped

- `ttp:Kill_Chains`
- any MAEC content

An Example

STIX 1.x in XML

```
<stix:TTP id="example:ttp-e610a4f1-9676-eab3-bcc6-b2768d58281a"
          xsi:type='ttp:TTPType'
          timestamp="2014-05-08T09:00:00.000000Z">
   <ttp:Title>Poison Ivy</ttp:Title>
   <ttp:Behavior>
       <ttp:Malware>
           <ttp:Malware_Instance id="example:malware-fdd60b30-b67c-11e3-b0b9-
↪f01faf20d111">
               <ttp:Type xsi:type="stixVocabs:MalwareTypeVocab-1.0">Remote Access␣
↪Trojan</ttp:Type>
               <ttp:Name>Poison Ivy</ttp:Name>
           </ttp:Malware_Instance>
       </ttp:Malware>
   </ttp:Behavior>
</stix:TTP>
```

STIX 2.x in JSON

```
{
   "created": "2017-01-27T13:49:53.997Z",
   "description": "\n\nTITLE:\n\tPoison Ivy",
   "id": "malware--fdd60b30-b67c-11e3-b0b9-f01faf20d111",
   "malware_types": [
       "remote-access-trojan"
   ],
   "modified": "2017-01-27T13:49:53.997Z",
   "name": "Poison Ivy",
   "spec_version": "2.1",
   "type": "malware"
}
```

`malware_types` would be `labels` in 2.0

4.12 Observed Data

The Observed Data object in STIX 2.x corresponds to the `Observable` object in CybOX 2.x. Each Observed Data object contains or references one or more *related* cyber observable objects.

STIX 2.x adds two properties: `first_observed` and `last_observed`. These properties are related to the `number_observed` property, because it is possible for Observed Data to indicate that either one, or multiple instances of the same cyber observable occurred. If the `number_observed` property is 1, then the `first_observed` and `last_observed` properties contain the same timestamp, otherwise they are the timestamp of the first and last times that cyber observable occurred.

The `sighting_count` property of STIX 1.x may seem to be the same concept as `number_observed` property, but because STIX 2.x has made explicit the difference between sightings and observed data, this is not the case. See the STIX 2.x specification for more details. The sightings count is captured on the `sighting` SRO.

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`sighting_count`	not to be confused with **number_observed**
`Keywords`	`labels`

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Object`	`objects` in 2.0, `object_refs` in 2.1

STIX 1.x Properties Mapped Using STIX 2.x Relationships

none

STIX 1.x Properties Handled Based on the "missing policy"

none

STIX 1.x Properties Not Mapped

- `negate`
- `Event`
- `Title`
- `Description`
- `Pattern_Fidelity`
- `Observable_Source`

An Example

STIX 1.x in XML

```
<cybox:Observable id="example:observable-c8c32b6e-2ea8-51c4-6446-7f5218072f27">
   <cybox:Object id="example:object-d7fcce87-0e98-4537-81bf-1e7ca9ad3734">
        <cybox:Properties xsi:type="FileObj:FileObjectType">
            <FileObj:File_Name>iprip32.dll</FileObj:File_Name>
            <FileObj:File_Path>/usr/local</FileObj:File_Path>
            <FileObj:Hashes>
                <cyboxCommon:Hash>
                    <cyboxCommon:Type condition="Equals" xsi:type=
↪"cyboxVocabs:HashNameVocab-1.0">SHA256</cyboxCommon:Type>
                    <cyboxCommon:Simple_Hash_Value condition="Equals">
↪e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855</
↪cyboxCommon:Simple_Hash_Value>
                </cyboxCommon:Hash>
            </FileObj:Hashes>
        </cybox:Properties>
   </cybox:Object>
</cybox:Observable>
```

STIX 2.0 in JSON

```
{
   "created": "2017-01-27T13:49:41.345Z",
   "first_observed": "2017-01-27T13:49:41.345Z",
   "id": "observed-data--c8c32b6e-2ea8-51c4-6446-7f5218072f27",
   "last_observed": "2017-01-27T13:49:41.345Z",
   "modified": "2017-01-27T13:49:41.345Z",
   "number_observed": 1,
   "objects": {
       "0": {
            "hashes": {
                "SHA-256":
↪"e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"
```

(continues on next page)

(continued from previous page)

```
             },
            "name": "iprip32.dll",
            "parent_directory_ref": "1",
            "type": "file"
        },
        "1": {
            "path": "/usr/local",
            "type": "directory"
        }
    },
    "type": "observed-data"
}
```

STIX 2.1 in JSON

```
{
    "hashes": {
        "SHA-256": "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"
    },
    "id": "file--49959589-27c4-5873-8e23-82f6c909d4ca",
    "name": "iprip32.dll",
    "parent_directory_ref": "directory--4aa982e3-4aac-5d5b-a699-d08c8c11f5f3",
    "type": "file"
}

{
    "id": "directory--4aa982e3-4aac-5d5b-a699-d08c8c11f5f3",
    "path": "/usr/local",
    "type": "directory"
}

{
        "created": "2017-01-27T13:49:41.345Z",
        "first_observed": "2017-01-27T13:49:41.345Z",
        "id": "observed-data--c8c32b6e-2ea8-51c4-6446-7f5218072f27",
        "last_observed": "2017-01-27T13:49:41.345Z",
        "modified": "2017-01-27T13:49:41.345Z",
        "number_observed": 1,
        "object_refs": [
             "directory--4aa982e3-4aac-5d5b-a699-d08c8c11f5f3",
             "file--49959589-27c4-5873-8e23-82f6c909d4ca"
        ],
        "spec_version": "2.1",
        "type": "observed-data"
}
```

In STIX 2.x cyber observables are only used within `observed-data` objects to represent something that has actually been seen. In STIX 1.x if an `Observable` is contained in an `Indicator`, it is instead expressing a pattern to match against observed data.

The pattern expression to match the example cyber observable, when it is located in an indicator object, would be:

```
[(file:hashes.'SHA-256' =
↪'e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855' AND (file:name =
↪'iprip32.dll' AND file:parent_directory_ref.path = '/usr/local'))]",
```

4.13 Report

The Report object in STIX 2.x does not contain objects, but only object references to STIX objects that are specified elsewhere (the location of the actual objects may not be contained in the same bundle that contains the `report` object).

In STIX 2.x, properties that were associated with the report header in STIX 1.x are located in the `report` object itself. The `labels` property (`report_type` in 2.1) contains vocabulary literals similar to the ones contain in the `Intent` property in STIX 1.x.

The `published` property is required in STIX 2.x, so the timestamp of the STIX 1.2 Report is used.

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

none

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Observables`	`object_refs`
`Indicators`	`object_refs`
`TTPs`	`object_refs`
`Exploit_Targets`	`object_refs`
`Courses_Of_Action`	`object_refs`
`Campaigns`	`object_refs`
`Threat_Actors`	`object_refs`
`Report:Header.Intent`	`labels` in 2.0, `report_types` in 2.1
`Report:Header.Description`	`description`
`Report:Header.Title`	`name`

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
`Related_Reports`	`related-to` (when not used for versioning)

An Example

STIX 1.x in XML

```
<stix:Report timestamp="2015-05-07T14:22:14.760467+00:00"
             id="example:Report-ab11f431-4b3b-457c-835f-59920625fe65"
             xsi:type='report:ReportType' version="1.0">
        <report:Header>
            <report:Title>Report on Adversary Alpha's Campaign against the Industrial␣
↪Control Sector</report:Title>
            <report:Intent xsi:type="stixVocabs:ReportIntentVocab-1.0">Campaign␣
↪Characterization</report:Intent>
            <report:Description>Adversary Alpha has a campaign against the ICS sector!
↪</report:Description>
        </report:Header>
        <report:Campaigns>
            <report:Campaign idref="example:campaign-1855cb8a-d96c-4859-a450-
↪abb1e7c061f2" xsi:type='campaign:CampaignType'/>
        </report:Campaigns>
    </stix:Report>
```

STIX 2.x in JSON

```
{
        "created": "2015-05-07T14:22:14.760Z",
        "created_by_ref": "identity--c1b58a86-e037-4069-814d-dd0bc75539e3",
        "description": "Adversary Alpha has a campaign against the ICS sector!
↪\n\nINTENT:\nCampaign Characterization",
        "id": "report--ab11f431-4b3b-457c-835f-59920625fe65",
        "report_types": [
            "campaign-characterization"
        ],
        "modified": "2015-05-07T14:22:14.760Z",
        "name": "Report on Adversary Alpha's Campaign against the Industrial Control␣
↪Sector",
        "object_refs": [
            "campaign--1855cb8a-d96c-4859-a450-abb1e7c061f2"
        ],
        "spec_version": "2.1",
        "type": "report"
    }
```

`report_types` would be `labels` in 2.0

4.14 Threat Actor

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Intended_Effects`	`goals`
`Type`	`labels` in 2.0, `threat_actor_types` in 2.1

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
`Motivation`	`primary_motivation`, `secondary_motivations`, `personal_motivations`
`Sophistication`	`sophistication`

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
`Identity`	`attributed-to`
`Observed_TTPs`	`uses`
`Associated_Campaigns`	`attributed-to` (reverse)
`Associated_Actors`	`related-to` (when not used for versioning)

STIX 1.x Properties Handled Based on the "missing policy"

- `Planning_And_Operational_Support`

STIX 1.x Properties Not Mapped

none

An Example

STIX 1.x in XML

```
<stix:Threat_Actor id="example:threatactor-56f3f0db-b5d5-431c-ae56-c18f02caf500"
                   xsi:type='ta:ThreatActorType'
                   timestamp="2016-08-08T15:50:10.983Z"
                   version="1.2">
     <ta:Title>Fake BPP (Branistan Peoples Party)</ta:Title>
     <ta:Identity id="example:Identity-8c6af861-7b20-41ef-9b59-6344fd872a8f">
        <stixCommon:Name>Franistan Intelligence</stixCommon:Name>
     </ta:Identity>
     <ta:Type>
        <stixCommon:Value xsi:type="stixVocabs:ThreatActorTypeVocab-1.0">State Actor /
↪ Agency</stixCommon:Value>
     </ta:Type>
     <ta:Intended_Effect>Influence the election in Branistan</ta:Intended_Effect>
     <ta:Motivation>
        <stixCommon:Value xsi:type="stixVocabs:MotivationVocab-1.1">Political</
↪stixCommon:Value>
     </ta:Motivation>
     <ta:Motivation>
        <stixCommon:Value xsi:type="stixVocabs:MotivationVocab-1.1">Ideological</
↪stixCommon:Value>
     </ta:Motivation>
     <ta:Motivation>
        <stixCommon:Value>Organizational Gain</stixCommon:Value>
     </ta:Motivation>
     <ta:Sophistication>
        <stixCommon:Value>Strategic</stixCommon:Value>
     </ta:Sophistication>
</stix:Threat_Actor>
```

STIX 2.x in JSON

```
{
      "type": "threat-actor",
      "id": "threat-actor--56f3f0db-b5d5-431c-ae56-c18f02caf500",
      "created_by_ref": "identity--f690c992-8e7d-4b9a-9303-3312616c0220",
      "created": "2016-08-08T15:50:10.983Z",
      "modified": "2016-08-08T15:50:10.983Z",
      "threat_actor_types": ["nation-state"],
      "goals": ["Influence the election in Branistan"],
      "primary_motivation": "political",
      "secondary_motivations": ["ideology", "organizational-gain"],
      "name": "Fake BPP (Branistan Peoples Party)",
      "sophistication": "strategic",
      "spec_version": "2.1"
}

{
      "type": "identity",
      "id": "identity--8c6af861-7b20-41ef-9b59-6344fd872a8f",
      "created_by_ref": "identity--f690c992-8e7d-4b9a-9303-3312616c0220",
      "created": "2016-08-08T15:50:10.983Z",
      "modified": "2016-08-08T15:50:10.983Z",
      "name": "Franistan Intelligence",
      "identity_class": "organization",
      "spec_version": "2.1"
}
```

(continues on next page)

(continued from previous page)

```
{
      "type": "relationship",
      "id": "relationship--5b271699-d2ad-468c-903d-304ad7a17d71",
      "created": "2016-08-08T15:50:10.983Z",
      "modified": "2016-08-08T15:50:10.983Z",
      "relationship_type": "attributed-to",
      "source_ref": "threat-actor--56f3f0db-b5d5-431c-ae56-c18f02caf500",
      "target_ref": "identity--8c6af861-7b20-41ef-9b59-6344fd872a8f"
}
```

threat_actor_types would be labels in 2.0

4.15 Tool

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
Name (from CybOX)	name
Type (from CybOX)	labels in 2.0, tool_types in 2.1
Description (from CybOX)	description
Version (from CybOX)	tool_version

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x property
ttp:Kill_Chain_Phases	kill_chain_phases
References (from CybOX)	external_references

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
ttp:Related_TTPs	uses (attack-pattern) (reverse), related-to (when not used for versioning), targets (identity)

STIX 1.x Properties Handled Based on the "missing policy"

- Vendor
- Service_Pack

STIX 1.x Properties Not Mapped

- Compensation_Model (from CybOX)
- Errors (from CybOX)
- Execution_Environment (from CybOX)
- ttp:Exploit_Targets
- ttp:Kill_Chains
- Metadata (from CybOX)

- Tool_Configuration (from CybOX)
- Tool_Hashes (from CybOX)
- Tool_Specific_Data (from CybOX)
- ttp:Victim_Targeting

An Example

STIX 1.x in XML

```
<stix:TTP id=example:tool--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f
          timestamp="2016-04-06T20:03:48.000Z">
  <ttp:Resources>
      <ttp:Tools>
         <ttp:Tool>
             <cyboxCommon:Name>VNCConnect</cyboxCommon:Name>
             <cyboxCommon:Type>remote-access</cyboxCommon:Name>
             <cyboxCommon:Vendor>RealVNC Ltd</cyboxCommon:Vendor>
             <cyboxCommon:Version>6.03</cyboxCommon:Version>
         </ttp:Tool>
     </ttp:Tools>
  </ttp:Resources>
</stix:ttp>
```

STIX 2.x in JSON

```
{
  "type": "tool",
  "id": "tool--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
  "created": "2016-04-06T20:03:48.000Z",
  "modified": "2016-04-06T20:03:48.000Z",
  "spec_version": "2.1",
  "tool_types": [ "remote-access"],
  "version": "6.03",
  "name": "VNCConnect"
}
```

tool_types would be labels in 2.0

4.16 Vulnerability

STIX 1.x Properties Mapped Directly to STIX 2.x Properties

none

STIX 1.x Properties Translated to STIX 2.x Properties

STIX 1.x property	STIX 2.x mapping
CVE_ID	external_references
OSVDB_ID	external_references
References	external_references

STIX 1.x Properties Mapped Using STIX 2.x Relationships

STIX 1.x property	STIX 2.x relationship type
et:Potential_COAs	mitigates
et:Related_Exploit_Targets	related-to (when not used for versioning)

STIX 1.x Properties Handled Based on the "missing policy"

- Discovered_DateTime
- Published_DateTime
- Source

STIX 1.x Properties Not Mapped

- is_known
- is_publicly_acknowledged
- CVSS_Score
- Affected_Software

An Example

STIX 1.x in XML

```
<stix:Exploit_Targets>
   <stixCommon:Exploit_Target id="example:et-e77c1e36-5b43-4c5c-b8cb-7b36035f2b90"␣
↪timestamp="2014-06-20T15:16:56.986650+00:00" xsi:type='et:ExploitTargetType'␣
↪version="1.2">
       <et:Title>Heartbleed</et:Title>
       <et:Vulnerability>
           <et:CVE_ID>CVE-2013-3893</et:CVE_ID>
       </et:Vulnerability>
   </stixCommon:Exploit_Target>
</stix:Exploit_Targets>
```

STIX 2.x in JSON

```
{
   "created": "2014-06-20T15:16:56.986Z",
   "external_references": [
       {
           "external_id": "CVE-2013-3893",
           "source_name": "cve"
       }
   ],
   "id": "vulnerability--e77c1e36-5b43-4c5c-b8cb-7b36035f2b90",
   "modified": "2017-01-27T13:49:54.310Z",
   "name": "Heartbleed",
   "type": "vulnerability"
}
```

CHAPTER 5

Mappings from CybOX 2.x to STIX 2.x

The following table associates the CybOX 2.x object types with their STIX 2.x cyber observable types. For each CybOX object the table also indicates if the elevator is able to convert the CybOX object to STIX 2.x.

CybOX object types not listed have no corresponding STIX 2.x cyber observable type, and therefore are not converted by the elevator.

Cybox 2.x Object Type	STIX 2.x Cyber Observable Type	Converted in the current
Address	email-addr	yes
Address	ipv4-addr	yes
Address	ipv6-addr	yes
Address	mac-addr	yes
ArchiveFile	file:archive-ext	yes
Artifact	artifact	yes
AutonomousSystem	autonomous-system	yes
File	directory	yes
DomainName	domain-name	yes
DSN Query	*none*	no
EmailMessage	email-message	yes
File*	file	yes
Hostname	domain-name	yes
HTTPClientRequest	network-traffic:http-request-ext	yes
HTTPSession	network-traffic	yes
ICMP (v4/v6)	network-traffic:icmp-ext	yes
ImageFile	file:raster-image-ext	yes
Link	*none*	no
Mutex	mutex	yes
NetworkConnection	network-traffic	yes
NetworkSocket	network-traffic:socket-ext	yes
PDFFile	file:pdf-ext	yes
Process*	process	yes
Product	software	yes

Table 1 – continued from previous page

Cybox 2.x Object Type	STIX 2.x Cyber Observable Type	Converted in the current
SocketAddress	network-traffic	yes
Hostname	domain-name	yes
Port	integer	yes
TCP	network-traffic:tcp-ext	no
URI	url	yes
UnixUserAccount	user-account:unix-account-ext	yes
UserAccount/WinUserAccount	user-account	yes
WindowsRegistryKey	window-registry-key	yes
WinExecutableFile	file:window-pebinary-ext	yes
WinFile	file:ntfs-ext	no
WinProcess	process:windows-process-ext	yes
WinService	process:windows-service-ext	yes
X509Certificate	x509-certificate	yes
X509V3Extensions	x509-certificate:x509-v3-extensions-type	yes

- Window or Unix Cybox object types handled by the basic STIX object type

5.1 CybOX 2.1 Object Types Not Representable in STIX 2.x

STIX 2.x can support these CybOX object types using Custom object (deprecated) or Extensions, but this is beyond the current scope of the Elevator.

- API
- ARP
- Code
- DNS Cache
- DNS Query
- DNS Record
- Device
- Disk Partition
- GUI Dialogbox
- GUI
- GUI Window
- Library
- Link
- Linux Package
- Memory
- Network Flow
- Network Packet
- Network Route Entry/Unix Network Route Entry/Win Network Route Entry
- Network Route

- Network Subnet
- Pipe/Unix Pipe/Win Pipe
- SMS Message
- Semaphore/Win Semaphore
- System/Win System
- URL History
- User Session
- Volume/Unix Volume/Win Volume
- Whois
- Win Critical Section
- Win Driver
- Win Event Log
- Win Event
- Win Filemapping
- Win Handle
- Win Hook/Win Kernel Hook
- Win Kernel
- Win Mailslot
- Win Memory Page Region
- Win Network Share
- Win Prefetch
- Win System Restore
- Win Task
- Win Thread
- Win Waitable Timer

5.2 Converting Network Cyber Observables

Most of the mappings between CybOX 2.x objects and STIX 2.x cyber observables are straightforward, therefore, they will not be detailed in this document. However, it would be advantageous to detail the mappings of network-traffic, a "catch-all" STIX 2.x cyber observable type for information previously represented in CybOX 2.x by:

- NetworkConnection
- HTTPSessionObject
- NetworkFlowObject
- NetworkPacket

This information is organized very differently than in CybOX 2.x. In addition, many CybOX 2.x properties are not available in the `network-traffic` object.

When converting network cyber observables, the elevator will often infer entries of the `protocols` property.

Notice that although both STIX 1.x and 2.x have object types to represent TCP packets, they are not compatible, so no conversion is made.

CybOX 2.x Type	STIX 2.0 mapping
`NetworkConnection`	`network-traffic`
`HTTPSessionObject/HTTPSessionObject/HTTPClientRequest`	`network-traffic/http-request-ext`
`NetworkFlowObject/UnidirectionalRecord/IPFIXMessage`	`network-traffic/ipfix`
`NetworkPacket/InternetLayer/ICMPv(4/6)`	`network-traffic/icmp-ext`
`NetworkSocket`	`network-traffic/socket-ext`

CHAPTER 6

Vocabularies

In STIX 2.x, vocabularies are referred to as "open". Although vocabularies in STIX 1.x were referred to as "controlled", the actual difference between them is negligible. In both standards, vocabulary literals were suggested, but not required to be used. Producers using either standards are free to use any string as a value. The most important difference is that in STIX 1.x it was possible to require that only suggested literals were used, and have that enforced through XML schema validation.

Certain STIX 2.x vocabularies are either copied verbatim from STIX 1.x, or with few changes. Others, are revamped in STIX 2.x, and it might be difficult to find a corresponding literal to one from STIX 1.x. However, because all of these vocabularies are open in STIX 2.x, those values can be used directly.

STIX 1.x Vocabulary	STIX 2.x Vocabulary
`AssetTypeVocab`	*not available in STIX 2.x*
`AttackerInfrastructureTypeVocab`	*not available in STIX 2.x*
`AttackerToolTypeVocab`	`tool-label-ov` (2.0) `tool-type-ov` (2.1)
`AvailabilityLossTypeVocab`	*not available in STIX 2.x*
`COAStageVocab`	*not available in STIX 2.x*
`CampaignStatusVocab`	*not available in STIX 2.x*
`CourseOfActionTypeVocab`	*not available in STIX 2.x*
`DiscoveryMethodVocab`	*not available in STIX 2.x*
`HighMediumLowVocab`	*not used*
`ImpactQualificationVocab`	*not available in STIX 2.x*
`ImpactRatingVocab`	*not available in STIX 2.x*
`IncidentCategoryVocab`	*not available in STIX 2.x*
`IncidentEffectVocab`	*not available in STIX 2.x*
`IncidentStatusVocab`	*not available in STIX 2.x*
`IndicatorTypeVocab`	`indicator-label-ov` (2.0) `indicator-type-ov` (2.1)
`InformationSourceRoleVocab`	*not available in STIX 2.x*
`InformationTypeVocab`	*not available in STIX 2.x*
`IntendedEffectVocab`	*not available in STIX 2.x*
`LocationClassVocab`	*not available in STIX 2.x*
`LossDurationVocab`	*not available in STIX 2.x*

Continued on next page

Table 1 – continued from previous page

STIX 1.x Vocabulary	STIX 2.x Vocabulary
`LossPropertyVocab`	*not available in STIX 2.x*
`MalwareTypeVocab`	`malware-label-ov` (2.0) `malware-type-ov` (2.1)
`ManagementClassVocab`	*not available in STIX 2.x*
`MotivationVocab`	`attack-motivation-ov`
`OwnershipClassVocab`	*not available in STIX 2.x*
`PackageIntentVocab`	*not used*
`PlanningAndOperationalSupportVocab`	`attack-resource-level-ov`
`ReportIntentVocab`	`report-label-ov` (2.0) `report-type-ov` (2.1)
`SecurityCompromiseVocab`	*not used*
`SystemTypeVocab`	*not available in STIX 2.x*
`ThreatActorSophisticationVocab`	`threat-actor-sophistication-level-ov`
`ThreatActorTypeVocab`	`threat-actor-label-ov` (2.0) `threat-actor-type-ov` (2.1)
`VersioningVocab`	*not used*

New vocabularies added in STIX 2.x are:

- `attack-resource-level-ov`
- `encryption-algo-ov`
- `extension-type-ov`
- `grouping-context-ov`
- `hash-algorithm-ov`
- `identity-class-ov`
- `implementation-language-ov`
- `industry-sector-ov`
- `infrastructure-type-ov`
- `malware-result-ov`
- `malware-capabilities-ov`
- `pattern-type-ov`
- `threat-actor-role-ov`
- `processor-architecture-ov`
- `region-ov`
- `threat-actor-role-ov`
- `windows-pebinary-type-ov`

In addition, the STIX 2.x specification contains enumerations. These are mostly for cyber observables. These are different from open vocabularies because only values explicitly defined in the enumeration can be used. The enumerations defined in STIX 2.x are:

- `network-socket-type-enum`
- `network-socket-address-family-enum`
- `opinion-enum`
- `windows-integrity-level-enum`
- `windows-registry-datatype-enum`

- `windows-service-start-type-enum`
- `windows-service-status-enum`
- `windows-service-type-enum`

which correspond to similar enumerations defined in STIX 1.x.

CHAPTER 7

Conversion Issues

This section discusses some techniques to facilitate the conversion of STIX 1.x data to STIX 2.x. These techniques cover non-obvious issues that might present an impediment to re-using STIX 1.x data.

7.1 Assumptions

7.1.1 Timestamps, Identifiers and Object Creators

In STIX 1.x most properties were optional. This includes properties that correspond to required properties in STIX 2.x. In particular, all STIX SDOs, SMOs and SROs in 2.x are required to have `id` and `created` properties. In STIX 2.1, all SCOs must have the `id` property. These are often not specified in a STIX 1.x object, but can sometimes be inferred from another STIX 1.x object in the same package.

Content in STIX 1.x was often hierarchical unlike content in STIX 2.x which is relatively flat, and this can help to determine required properties. For instance, a timestamp on a STIX 1.x package could be construed as the timestamp for all objects it contains. Likewise, an object could assume that its parent object's timestamp is also the timestamp of that object, unless that object possessed its own timestamp. Of course, if no timestamp is present for any of the objects, included the top level package, some other timestamp outside of the content must be used. In most cases, this would probably result in using the current timestamp when the conversion is made.

Most top-level STIX 1.x objects contained an `id` (or an `idref`), however when converting STIX 1.x TTPs and Exploit Targets the id must be assigned to the STIX 2.x object that results. For instance, a TTP might have contain an attack pattern object, but the id was not a property of the attack pattern, but the TTP.

In certain circumstances, no id is available or in the case of TTPs and Exploit Targets, there may be more than one STIX 2.x object created. In these cases, a new `id` must be used.

In STIX 1.x, all top-level objects had a `Information_Source` property to hold information about, among other things, the object creator. However, this property was optional. `created_by_ref`, which is a common property on all STIX 2.x SDOs, SMOs and SROs, is often optional. It should be noted however, that the object creator can also be "inherited" from its parent object, as with the timestamp. This fact can be useful to derive a more robust STIX 2.x object. Note that SCOs do not have a `created_by_ref` property.

7.1.2 Special Considerations for TTPs and Exploit Target Conversions

When converting a STIX 1.x TTP or Exploit Target certain properties exist at the top-level, and not in the subsidiary object which will form the basis of the STIX 2.x object. However, those properties must be used when creating the subsidiary object. See section *Attack Pattern* for an example. The conversion of that STIX 1.x TTP will yield a STIX 2.x Attack Pattern, whose `name` and `created_by_ref` are determined from the TTP itself, and not the STIX 1.x Attack Pattern.

7.1.3 Minor Issues

- The `condition` property was optional in STIX 1.x Observables. If it was not specified for an Observable used for patterning, the condition used in the STIX 2.x pattern will be assumed to be "=".
- The title property should be used for the `name` property, when necessary.
- STIX 1.2 introduced versioning of objects. Currently, there is no guidance to converting STIX 1.2 versioning to STIX 2.x versioning. In most cases, a STIX 1.x relationship between object instances of the same type will be converted to a `related-to` relationship in STIX 2.x, which could be undesirable.

7.2 Optional vs. Required

Certain properties are required in STIX 2.x object that were optional in STIX 1.x. This goes beyond the properties such as ids, created/modified timestamps. The most frequently occurring example is the `labels` property in 2.0. The elevator will use a default value - `unknown`. Other SDOs have similarly named properties.

7.3 Issues with Patterns

Patterns in STIX 2.x have certain restrictions that didn't explicitly appear in STIX 1.x. A pattern in STIX 2.x has explicit rules about if the expression can refer to only one or many observed data instances. Because STIX 1.x patterns did not have any of these restrictions, a reasonable conversion of the pattern by the elevator might be illegal in STIX 2.x.

Additionally, the use of the NOT operator in STIX 2.x is restricted to be used only with Comparison operators. Therefore, it is not possible to express a pattern such as `NOT (file.name == foo.bar" AND 'file.size == 123)` directly. To yield an equivalent pattern expression in STIX 2.x, DeMorgan's Law would need to be used to reduce the scope of the NOT operator: `(file.name != foo.bar" OR 'file.size != 123)`, but the elevator does not perform this functionality.

7.4 Single vs. Multiple

Some properties in STIX 1.x allowed for multiple values, but the corresponding property in STIX 2.x does not. In these cases, the first value is used.

In certain situations, something specific to the properties can be helpful in handling this issue. For instance, the first entry in the STIX 1.x Threat Actors `motivation` property should be assumed to be the `primary_motivation`. Any others should be listed in the `secondary_motivations` property.

7.5 Data Markings

The stix-elevator currently supports global markings and object-level markings. Through the use of hashing, the elevator make the best effort to detect duplicate markings to prevent excessive object creation. Also, the marking types supported by the elevator is limited to: Simple, Terms of Use, TLP and AIS. AIS is a data marking used when submitting STIX content to DHS/CISA.

7.6 Missing Policy

Certain STIX 1.x properties cannot be converted to a STIX 2.x property defined in the STIX 2.x specification. The elevator provides a command line option to determine how to handle these STIX 1.x properties.

- `add-to-description`: Add the property name, property value pair to the description property to the `description` property.
- `use-custom-properties`: STIX 2.x provides the ability to add *custom* properties to any STIX object. Missing properties can be included using this facility. Note, that custom property names will have a prefix of `x_<CUSTOM_PROPERTY_PREFIX>`, where `CUSTOM_PROPERTY_PREFIX` is provided as a command line option. It defaults to `elevator`. This option has been deprecated, use `use-extensions` instead.
- `use-extensions`: STIX 2.x provides the ability to "extend" any STIX object, using the extension-definition object.
- `ignore`: The content is dropped, and does not appear in the STIX 2.x object

Note that the handling of missing properties is not complete - not every STIX 1.x property is handled. The Mapping section of this documentation lists what properties are handled for each SDO.

The disposition of all missing properties is presented in warning messages.

It is possible to create custom cyber observables in STIX 1.x through use of the CustomObjectType. This can only be done within an Observable Object, therefore the resulting STIX 2.1 object will be a SCO. For STIX 2.0, it will be similar to any other cyber observable object.

`Incident` and `Infrastructure` are object types in STIX 1.x, but it is not representable in STIX 2.0. However, through the use of the options –incidents and –infrastructure, a custom object (or extensions) will be created. Both of these object types exist in STIX 2.1.

An Example

STIX 1.x

```
<stix:Course_Of_Action id="example:coa-495c9b28-b5d8-41e3-b7bb-000c29789db9" xsi:type=
↪'coa:CourseOfActionType' version="1.2">
        <coa:Title>Block outbound traffic</coa:Title>
        <coa:Stage xsi:type="stixVocabs:COAStageVocab-1.0">Response</coa:Stage>
        <coa:Type xsi:type="stixVocabs:CourseOfActionTypeVocab-1.0">Perimeter Blocking
↪</coa:Type>
        <coa:Objective>
            <coa:Description>Block communication between the PIVY agents and the C2␣
↪Server</coa:Description>
            <coa:Applicability_Confidence>
                <stixCommon:Value xsi:type="stixVocabs:HighMediumLowVocab-1.0">High</
↪stixCommon:Value>
            </coa:Applicability_Confidence>
        </coa:Objective>
        <coa:Impact>
```

(continues on next page)

(continued from previous page)

```
            <stixCommon:Value xsi:type="stixVocabs:HighMediumLowVocab-1.0">Low</
↪stixCommon:Value>
            <stixCommon:Description>This IP address is not used for legitimate␣
↪hosting so there should be no operational impact.</stixCommon:Description>
        </coa:Impact>
    </stix:Course_Of_Action>
```

STIX 2.x using add-to-description

```
{
        "created": "2015-07-31T11:24:39.090Z",
        "description": "\n\nSTAGE:\n\tResponse\n\nOBJECTIVE: Block outbound␣
↪traffic\n\nOBJECTIVE CONFIDENCE: High\n\nIMPACT:Medium: Some description about the␣
↪indicator.",
        "id": "course-of-action--3dbfccad-1fbb-4e9f-8307-f2d1a5c651cc",
        "labels": [
            "perimeter-blocking"
        ],
        "modified": "2015-07-31T11:24:39.090Z",
        "name": "Block outbound traffic",
        "spec_version": "2.1",
        "type": "course-of-action"
}
```

STIX 2.x using use-extensions

```
{
        "created": "2015-07-31T11:24:39.090Z",
        "extensions": {
            "extension-definition--a46b18de-0b41-4a95-9d2d-67a360f2d859": {
                "extension_type": "property-extension",
                "impact": {
                    "description": "Some description about the indicator.",
                    "value": "Medium"
                },
                "objective": "Block outbound traffic",
                "objective_confidence": "High",
                "stage": "Response"
            }
        },
        "id": "course-of-action--3dbfccad-1fbb-4e9f-8307-f2d1a5c651cc",
        "labels": [
            "perimeter-blocking"
        ],
        "modified": "2015-07-31T11:24:39.090Z",
        "name": "Block outbound traffic",
        "spec_version": "2.1",
        "type": "course-of-action"
}
```

STIX 2.x using use-custom-properties

```
{
    "created": "2015-07-31T11:24:39.090Z",
    "id": "course-of-action--3dbfccad-1fbb-4e9f-8307-f2d1a5c651cc",
    "labels": [
        "perimeter-blocking"
```

(continues on next page)

(continued from previous page)

```
    ],
    "modified": "2015-07-31T11:24:39.090Z",
    "name": "Block outbound traffic",
    "spec_version": "2.1",
    "type": "course-of-action",
    "x_elevator_impact": {
        "description": "Some description about the indicator.",
        "value": "Medium"
    },
    "x_elevator_objective": "Block outbound traffic",
    "x_elevator_objective_confidence": "High",
    "x_elevator_stage": "Response"
}
```

STIX 2.x using `ignore`

```
{
    "created": "2015-07-31T11:24:39.090Z",
    "id": "course-of-action--3dbfccad-1fbb-4e9f-8307-f2d1a5c651cc",
    "labels": [
        "perimeter-blocking"
    ],
    "modified": "2015-07-31T11:24:39.090Z",
    "name": "Block outbound traffic",
    "type": "course-of-action"
}
```

7.7 Extensions

Extensions are based on the Extension Definition object. The key in the `extension` property dictionary contains the id of the Extension Definition object used to define the extension. Extensions are explained in detail in section 7.3 of the STIX 2.1 specification document.

Currently, the schemas associated with the Extension Definition object do not exist. However, the Extension Definition objects themselves can be found in extension_definitions.py. They will be more fully defined in a future release of the elevator.

Note that these extensions are not used by the predefined extension (e.g., Archive File), because those are fully defined within the specification.

CHAPTER 8

Warning Messages

When the elevator makes an assumption during the conversion of some content, or is unable to convert the content, a warning message is output.

8.1 General

Message	Code	Level
Results produced by the stix2-elevator may generate warning messages which should be investigated	201	info
Observable Expressions should not contain placeholders	202	error
Placeholder *id* should be resolved	203	error
Found definition for *id*	204	info
At least one PLACEHOLDER idref was not resolved in *id*	205	error
At least one observable could not be converted in *id*	206	error
Options not initialized	207	error
EMPTY BUNDLE – No objects created from 1.x input document!	208	warn
Both console and output log have disabled messages.	209	info
OSError *message*	210	error
silent option is not compatible with a policy	211	warn
Created Marking Structure for *id*	212	info
custom_property_prefix is provided, but the missing policy is not 'use-custom-properies'. It will be ignored.	213	warn
type option was not given, but it defaults to true for version 2.1"	214	info
Custom properties/objects/extensions are deprecated in version 2.1. Suggest using 'use-extensions' instead	215	info
The missing policy option of 'use-extensions' cannot be used with version 2.0. 'use-custom-properies' is suggested	216	error
ACS data markings cannot be supported in version 2.0. –acs option is ignored.	217	warn

8.2 Handle STIX 1.x Content not supported in STIX 2.x

Message	Code	Level
The `Short_Description` property is no longer supported in STIX. The text was appended to the description property of *id*	301	info
Appended *property_name* to description of *id*	302	warn
Title in *type* used for `name`, appending `exploit_target` *id* title in description property	303	info
Appended `confidence` property content to description of *id*	304	warn
Appended `Statement` type content to description of *id*	305	warn
Appended `Tool` type content to description of *id*	306	warn
Missing property *property_name* of *id* is ignored	307	warn
Used custom property for *property_name* of *id*	308	warn
Missing property *property_name* of *id* is ignored, because there is no description property	309	warn
The Short_Description property in *id* is not supported in STIX 2.x.	310	info
Used an extension for objective of *id*	311	warn
No extension-definition was found for STIX 1 type *type* in *id*	312	warn
Used extension property for *property_name* of *id*	313	warn
Property *property_name* of *id* is ignored, because it can't be represented in an extension	314	warn
New extension-definition id *id* was generated for *type*. *id*	315	warn
Custom Content *property_name* of *id* is ignored	316	warn
Used *object_path* for extension property for *property_name*	317	warn
Token in control set not recognized: *token*	318	warn
Used extensions for ACS data markings. See *id*	319	warn

8.3 Content not supported in STIX 2.x

Message
`Information Source` on *id* is not representable in STIX 2.x
`Related_Packages` type in *id* not supported in STIX 2.x
`Campaign/Activity` type in *id* not supported in STIX 2.x
Structured COAs type in *id* are not supported in STIX 2.x
`ExploitTarget/Weaknesses` type in *id* not supported in STIX 2.x
`ExploitTarget/Configurations` type in *id* not supported in STIX 2.x
Indicator *id* has an observable or indicator composite expression which may not supported correctly in STIX 2.x - please check this pa
`TTP/Behavior/Exploits/Exploit` in *id* not supported in STIX 2.x
`Infrastructure` in *id* not part of STIX 2.0
IOC indicator in *id* cannot be converted to a STIX pattern
Relationship *rel_name* in *id* for *id* is not explicitly supported in STIX 2.x. Expression *pattern* is ANDed
Relationship *rel_name* in *id* for *id* is not explicitly supported in STIX 2.x. %s will be ANDed if/when resolved
Kill Chains type in *id* not supported in STIX 2.x
Victim Target in *id* did not yield any STIX 2.x object
TTP *id* did not generate any STIX 2.x object
No STIX 2.x object generated from embedded object *id*
object did not yield any STIX 2.x object
The *property* property of *STIX 1.x object type* is not part of STIX 2.x
id is used as a characteristic in an infrastructure object, therefore it is not included as an observed_data instance
Windows Handles are not a part of STIX 2.x
The address type address is not part of STIX 2.x

Co

Table 1 – continued from previous page

Message
No pattern term was created from *id*
id is used as a pattern, therefore it is not included as an observed_data instance
xxx content is not supported in STIX 2.x
Could not resolve Marking Structure *id*
MAEC content in *id* cannot be represented in STIX 2.x
The *relationship name* relationship involving *id* is not explicitly supported in STIX 2.x
`roles` is not a property of a 2.x identity (*id*). Perhaps the roles are associated with a related Threat Actor
`HTTPServerResponse` type is not supported in STIX 2.x
The confidence value *value* is not found on one of the confidence scales from the specification. No confidence can be inferred
The confidence value *value* is not between 0 and 100, which is required for STIX 2.1. No confidence can be inferred
The confidence value *value* cannot be converted
Location with free text address in *id* not handled yet
Observed Data objects cannot refer to other external objects: *property name* in *type*”
CIQ Address information in *id* is not representable in 2.0
ACS data markings only supported when –acs option is used. See *id*

8.4 Multiple values are not supported in STIX 2.x

Message	Code	Level
Cannot convert range of *ip addr 1* to *ip addr 2* in *id* to a CIDR	501	warn
Only one person name allowed for *id* in STIX 2.x, used *name_1*, *name_2* becomes an alias	502	warn
Only one organization name allowed for *id* in STIX 2.x, used *name_1*, *name_2* becomes an alias	503	warn
YARA/SNORT/IOC or other patterns are not supported in STIX 2.0. See *id*	504	warn
Only two pdfids are allowed for *id*, dropping *pidid*	505	warn
Only one alternative test mechanism allowed for *id* in STIX 2.x - used *pattern_lang_1*, dropped *pattern_lang_2*	506	warn
Only one valid time window allowed for *id* in STIX 2.x - used first one	507	warn
Only one name for malware is allowed for *id* in STIX 2.x - used *name_1*, dropped *name_2*	508	warn
No STIX 1.x vocab value given for *property*, using ‘unknown’	509	warn
Only one *property name* allowed in STIX 2.x - used *prop_value* in *id*	510	warn
File size ‘window’ not allowed in top level observable, using first value	511	warn
Only one `HTTP_Request_Response` used for `http-request-ext`, using first value	512	warn

8.5 Possible issue in original STIX 1.x content

Message
Dangling source reference *source* in *id*
Dangling target reference *target* in *id*
STIX 1.X ID: *id* was not mapped to STIX 2.x ID
Unable to determine the STIX 2.x type for *id*
Malformed id *id*. Generated a new uuid
Identity *id* has organization and person names
Dangling kill chain phase id in indicator *id*
`windows-registry-key` is required to have a `key` property
condition was used, but two values were not provided.

Continu

Table 2 – continued from previous page

Message
No object mapped to *old_id*
Can not associate *old_id* with None
Identity *id* must have a name, using ‘None’
No *type* properties found in *object*
Address direction in *id* is inconsistent, using ‘src’”
No `WinProcess` properties found in *WinProcess*
No `WinService` properties found in *WinService*
The custom property name *name* does not adhere to the specification rules
No ISO code for *value* in *identifying_info*
No *start/end* time for the first valid time interval is available in *id*, other time intervals might be more appropriate
Unable to create a pattern from a File object
stix_1.x_property contains no value
No term was yielded for *id*
Hive property, *hive_property_name*, is already a prefix of the key property, *key property name*
The custom property name *name* contains whitespace, replacing it with underscores
Found duplicate marking structure *id*
hash_string is not a valid *hash_type* hash
enum_value in *id* is not a member of the *enum_type* enumeration
Unknown condition given in *id* - marked as ‘INVALID_CONDITION’
Unable to determine the STIX 2.x type for *id*, which is malformed
‘equals’ allowed in *id* - should be ‘Equals’
Multiple administrative areas with multiple countries in *id* is not handled
Unknown phase_id *phase_id* in *id*
File path directory is empty *file_path*
Any artifact additional artifact info on *id* is not recoverable
id contains a observable composition, which implies it not an observation, but a pattern and needs to be contained within an indicator.
Address direction in *id* is not provided, using ‘src’
cisa-proprietary is only permitted when ais-consent is everyone, so it has been dropped. See *id*
Indicator *id* does not contain the information necessary to generate a pattern
This observable *id* already is associated with cyber observables
Unable to determine the hash type for *hash value*
Required property *property* is not provided for ACS data marking

8.6 STIX Elevator conversion based on assumptions

Message	Code	Level
Threat Actor identity *id* being used as basis of attributed-to relationship	701	info
Found STIX 1.X ID: *old_id* replaced by *new_id*	702	info
old_id is already associated other ids: *tuple_of_new_ids*	703	info
Including *id of relationship* in *id of report* and added the target_ref *target_ref* to the report	704	warn
Including *id of relationship* in *id of report* and added the source_ref *source_ref* to the report	705	warn
Including *id of relationship* in *id of report* although the target_ref is unknown	706	warn
Including *id of relationship* in *id of report* although the source_ref is unknown	707	warn
Not including *id of relationship* in *id of report* because there is no corresponding SDO for *target_ref*	708	warn
Not including *id of relationship* in *id of report* because there is no corresponding SDO for *source_ref*	709	warn
All associated *relationship name* relationships of *id* are assumed to not represent STIX 1.2 versioning	710	info
ciq name found in *id*, possibly overriding other name	711	warn
Only one type pattern can be specified in *id* - using 'stix'	712	warn
id generated an identity associated with a victim	713	info
No condition given for term in *current_observable* - assume '='	714	warn
Used MATCHES operator for *condition*	715	info
Based on CIQ information, *id* is assumed to be an organization	716	warn
Threat actor *id* title is used for name property	717	info
Using *relationship_name* for the *property* of *id*	718	warn
Using first Threat Actor motivation as `primary_motivation` value. If more, use `secondary_motivation`	719	info
The `published property` is required for STIX 2.x Report *id*, using the created property	720	info
`apply_condition` assumed to be 'ANY' in *id*	721	warn
`content_type` for `body_multipart` of attachment *id* is assumed to be 'text/plain'	722	info
The confidence value in *value* assumed to be a value on a scale between 0 and 100	723	warn
The confidence value in *value* has been converted to an integer so it is valid in STIX 2.1	724	warn
port number is assumed to be a destination port	725	warn
`Not in use`	726	warn
Custom property name *property* has been converted to all lower case	727	warn
The is_family property of malware instance *id* is assumed to be true	728	info
Included parent markings for Relationship *id* and Location *id*	729	info

8.7 STIX elevator currently doesn't process this content

Message	Code	Level
Could not resolve Marking Structure *id*	801	warn
STIX 1.x full file paths are not processed, yet	802	warn
Location *id* may not contain all aspects of the STIX 1.x CIQAddress object	803	warn
Object reference *id* may not be handled correctly	804	warn
CybOX object *object* not handled yet	805	warn
Email *property* not handled yet	806	warn
`file:extended_properties:windows_pebinary_ext:optional_header` is not implemented yet	807	warn
object found in *id* cannot be converted to a pattern, yet.	808	warn
Related Objects of cyber observables for *id* are not handled yet. `Not currently in use.`	809	warn
Negation of *id* is not handled yet	810	warn
Custom object with no name cannot be handled yet	811	warn
Condition *condition* on a hive property not handled.	812	warn
Cannot convert CybOX 2.x class name *name* to an object_path_root_name	813	error
`Not in use`	814	warn
property in *id* are not handled, yet.	815	info
Ambiguous file path *path* was not processed	816	warn
Pattern expression with STIX 1.x custom objects in *id* is ignored	817	warn
Pattern expression with STIX 1.x custom properties in *id* is ignored	818	warn

8.8 Missing Required Timestamp

Message	Code	Level
`first_observed` and `last_observed` properties not available directly on *id* - using timestamp	901	info
Using parent object timestamp on *identifying_info*	902	info
No valid time position information available in *id*, using parent timestamp	903	warn
No `first_seen` property on *id* - using timestamp	904	info
Timestamp not available for *entity*, using current time	905	warn

CHAPTER 9

Contributing

We’re thrilled that you’re interested in contributing to the stix2-elevator! Here are some things you should know:

- contribution-guide.org has great ideas for contributing to any open-source project (not just this one).
- All contributors must sign a Contributor License Agreement. See CONTRIBUTING.md in the project repository for specifics.
- If you are planning to implement a major feature (vs. fixing a bug), please discuss with a project maintainer first to ensure you aren’t duplicating the work of someone else, and that the feature is likely to be accepted.

Now, let’s get started!

9.1 Setting up a development environment

We recommend using a virtualenv.

1. Clone the repository. If you’re planning to make pull request, you should fork the repository on GitHub and clone your fork instead of the main repo:

```
$ git clone https://github.com/yourusername/cti-stix-elevator.git
```

2. Install develoment-related dependencies:

```
$ cd cti-stix-elevator
$ pip install -r requirements.txt
```

3. Install pre-commit git hooks:

```
$ pre-commit install
```

At this point you should be able to make changes to the code.

9.2 Code style

All code should follow PEP 8. We allow for line lengths up to 160 characters, but any lines over 80 characters should be the exception rather than the rule. PEP 8 conformance will be tested automatically by Tox and Travis-CI (see below).

9.3 Testing

Note: All of the tools mentioned in this section are installed when you run `pip install -r requirements.txt`.

This project uses pytest for testing. We encourage the use of test-driven development (TDD), where you write (failing) tests that demonstrate a bug or proposed new feature before writing code that fixes the bug or implements the features. Any code contributions should come with new or updated tests.

Tests are created by creating a STIX 1.x file containing the content which will cause the elevator to execute the code you are testing. This file should be placed in the idioms-xml directory. Use the elevator command line to create json "golden" files - which contain the correct result you expect from the elevator. You should provide golden files for each version and missing property option. These files should be placed in the idioms-json-2.x-<missing-property option> directory.

Note: the number of test files must be the same across the idioms directories, using the same file names.

Running tests can be done using tox, discussed below.

tox allows you to test a package across multiple versions of Python. Setting up multiple Python environments is beyond the scope of this guide, but feel free to ask for help setting them up. Tox should be run from the root directory of the project:

```
$ tox
```

We aim for high test coverage, using the coverage.py library. Though it's not an absolute requirement to maintain 100% coverage, all code contributions must be accompanied by tests. To run coverage and look for untested lines of code, run:

```
$ pytest --cov=stix2elevator
$ coverage html
```

then look at the resulting report in `htmlcov/index.html`.

All commits pushed to the `master` branch or submitted as a pull request are tested with Travis-CI automatically.

9.4 Adding a dependency

One of the pre-commit hooks we use in our develoment environment enforces a consistent ordering to imports. If you need to add a new library as a dependency please add it to the *known_third_party* section of *.isort.cfg* to make sure the import is sorted correctly.

CHAPTER 10

Indices and tables

- genindex
- modindex
- search