stix2-validator Documentation

Release 3.0.0

OASIS Open

Contents:

1	Installat	tion
2	Usage	5
	2.1 As	s A Script
	2.2 As	s A Library
		TIX 2 Versions
		dditional Schemas
3	Options	
4	Checkin	ng STIX Content
	4.1 M	Iandatory Checks - STIX 2.1
	4.2 O ₁	ptional Checks - STIX 2.1
5	Contrib	outing 23
	5.1 Se	etting up a development environment
		ode style
		esting
		dding a dependency
6	Indices	and tables

The STIX Validator checks that STIX JSON content conforms to the requirements specified in the latest STIX 2 specifications. In addition to checking conformance with the JSON schemas, the validator checks conformance with requirements that cannot be specified in JSON schema, as well as with established "best practices." This validator is non-normative; in cases of conflict with the STIX specification, the specification takes precedence.

The STIX 2 specification contains two types of requirements: mandatory "MUST" requirements, and recommended "SHOULD" best practice requirements. The validator checks documents against the "MUST" requirements using JSON schemas. Some of these mandatory requirements cannot be implemented in JSON Schema, however, so the validator uses Python functions to check them. The "SHOULD" requirements are all checked by Python functions, and options may be used to ignore some or all of these recommended "best practices."

The STIX Validator uses the stix2-patterns validator to check that Indicator patterns conform to the STIX Patterning language and only reference properties valid for the objects in the pattern.

The validator also color-codes its output to make it easier to tell at a glance whether validation passed.

Contents: 1

2 Contents:

CHAPTER 1

Installation

Note: The STIX 2 validator requires Python 2.7 or 3.4+.

The easiest way to install the STIX 2 validator is with pip:

```
$ pip install stix2-validator
```

Note that if you instead install it by cloning or downloading the repository, you will need to set up the submodules before you install it:

```
$ git clone https://github.com/oasis-open/cti-stix-validator.git
$ cd cti-stix-validator/
$ git submodule update --init --recursive
$ python setup.py install
```

CHAPTER 2

Usage

2.1 As A Script

The validator comes with a bundled script which you can use to validate a JSON file containing STIX content:

```
$ stix2_validator <stix_file.json>
```

2.2 As A Library

You can also use this library to integrate STIX validation into your own tools. You can validate a JSON file:

```
from stix2validator import validate_file, print_results

results = validate_file("stix_file.json")
print_results(results)
```

You can also validate a JSON string, and check if the input passed validation:

```
from stix2validator import validate_string, print_results

stix_json_string = "..."
results = validate_string(stix_json_string)
if results.is_valid:
    print_results(results)
```

If your STIX is already in a Python dictionary (for example if you have already run json.loads()), use validate_instance() instead:

```
import json
from stix2validator import validate_instance, print_results
```

(continues on next page)

(continued from previous page)

```
stix_json_string = "..."
stix_obj = json.loads(stix_json_string)
results = validate_instance(stix_obj)
if results.is_valid:
    print_results(results)
```

You can pass a ValidationOptions object into validate_file(), validate_string(), or validate_instance() if you want behavior other than the default:

```
from stix2validator import ValidationOptions

options = ValidationOptions(strict=True)
results = validate_string(stix_json_string, options)
```

2.3 STIX 2 Versions

By default the validator will check content against the latest version of the STIX 2 specification. However, older versions can be checked with the version option. For example:

```
$ stix2_validator --version=2.0 <stix_file.json>
```

or in Python:

```
options = ValidationOptions(strict=True, version="2.0")
results = validate_string(stix_json_string, options)
```

2.4 Additional Schemas

The validator uses the STIX 2 JSON schemas as the basis for its validation, but you can also validate with your own additional schemas. This can help if you want to validate STIX content using custom objects, properties, or observables.

To do this use the --schema-dir argument:

```
$ stix2_validator --schema-dir /path/to/my/schemas <stix_file.json>
```

or in Python:

You can see some examples of custom schemas here. Note that if you want to add a custom property to an existing object type, your custom schema only needs to contain that property; the validator's built-in schemas are still checked against and will handle the rest.

6 Chapter 2. Usage

$\mathsf{CHAPTER}\,3$

Options

These are the different options that can be set, whether the validator is used as a command-line script or as a Python library. When using the validator as a library, these options can be passed as parameters to the ValidationOptions constructor.

Script	Library	Description	
FILES	files	A whitespace separated list of STIX files or directories of STIX files to validate.	
-r,recursive	recursi	Recursively descend into input directories.	
-s SCHEMA_DIR,	schema_o	Chustom schema directory. If provided, input will be validated against	
schemas		these schemas in addition to the STIX schemas bundled with this script.	
SCHEMA_DIR			
version	version	The version of the STIX specification to validate against (e.g. "2.0").	
-v,verbose	verbose	Print informational notes and more verbose error messages.	
-q,silent	silent	Silence all output to stdout.	
-d DISABLED,	disable	A comma-separated list of recommended best practice checks to skip.	
disable		By default, no checks are disabled. Example: –disable 202,210	
DISABLED,ignore			
DISABLED			
-e ENABLED,	enabled	1	
enable ENABLED,		If the –disable option is not used, no other checks will be run. By default,	
select ENABLED		all checks are enabled. Example: –enable 218	
strict	strict	Treat warnings as errors and fail validation if any are found.	
strict-types	strict_	Finesure that no custom object types are used, only those defined in the	
		STIX specification.	
strict-properties	strict_]	Finseretthetsno custom properties are used, only those defined in the	
		STIX specification.	
no-cache		Disable the caching of external source values.	
refresh-cache	refresh	Clears the cache of external source values, then during validation down-	
		loads them again.	
clear-cache		Chear the cache of external source values after validation.	
enforce-refs	enforce	Ensures that all SDOs being referenced by SROs are contained within	
		the same bundle.	

For the list of checks that can be used with the "enabled" or "disabled" options, see the *Best Practices page*.

CHAPTER 4

Checking STIX Content

The validator will always validate input against all of the mandatory "MUST" requirements from the spec. By default it will issue warnings if the input fails any of the "SHOULD" recommendations, but validation will still pass. To turn these "best practice" warnings into errors and cause validation to fail, use the --strict option with the command-line script, or create a ValidationOptions object with strict=True when using the library.

You cannot select which of the "MUST" requirement checks will be performed; all of them will always be performed. However, you may select which of the "SHOULD" checks to perform. Use the codes from the table below to enable or disable these checks. For example, to disable the checks for the report label and tool label vocabularies, use --disable 218,222 or disabled="218,222". All the other checks will still be performed. Conversely, to only check that custom property names adhere to the recommended format but not run any of the other "best practice" checks, use --enable 103 or enabled="103".

Enabling supersedes disabling. Simultaneously enabling and disabling the same check will result in the validator performing that check.

Some checks access Internet resources to determine valid values for certain properties. For instance, the 'mimetype' check accesses the IANA's list of registered MIME types. These web requests are cached to conserve bandwidth, will expire after one week, and are stored in a file called 'cache.sqlite' in the same directory the script is run from. The cache can be refreshed manually with the <code>--refresh-cache</code> or <code>refresh_cache=True</code>, or cleared with <code>--clear-cache</code> or <code>clear_cache=True</code>. This caching can be disabled entirely with <code>--no-cache</code> or <code>no-cache=True</code>.

4.1 Mandatory Checks - STIX 2.1

Name	e Ensures	Errors/Warnings
times	- timestamps contain sane	' <pre>'<pre>c '<pre>c '<pre>c</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre>
tamp	_	' <object>': '<pre>'<pre>'</pre> 's not a valid timestamp: <mes-< td=""></mes-<></pre></object>
1	utes, seconds	sage>
		' <object>': '<extension>': '<property>': '<timestamp>' is not a valid</timestamp></property></extension></object>
		timestamp: <message></message>
		' <object>': '<pre>'<pre>roperty>': '<embedded-property>' is not a valid timestamp:</embedded-property></pre></pre></object>
		<message></message>
times	- timestamp properties with a	' <operand_1>' (<operand1_value>) must be <comparison_op></comparison_op></operand1_value></operand_1>
tamp	ccompareison are valid	' <operand_2>' (<operand2_value)< td=""></operand2_value)<></operand_2>
ob-	cyber observable timestamp	In object ' <identifier>', '<operand_1>' (<operand1_value>) must be <com-< td=""></com-<></operand1_value></operand_1></identifier>
serv-	properties with a compari-	parison_op> ' <operand_2>' (<operand2_value>)</operand2_value></operand_2>
able_	ti soestæq w <u>ir</u> emeptaære valid	
ob-	that marking definitions do	'object_marking_refs' cannot contain any references to this marking defini-
ject_1	n and ting <u>o</u> kulackular <u>ir</u> cefkar refer-	tion object (no circular references)
	ences (i.e., they do not refer-	
	ence themselves in the 'ob-	
	ject_marking_refs' property	
gran-		'granular markings' cannot contain any references to this marking definition
u-	do not contain circular	object (no circular references)
lar_m	arkfogen_coiscu(ine_refsthey do	
	not reference themselves	
	in the 'granular_markings'	
	property	(destants to the control of the cont
	selectors in granular mark-	' <selector>' is not a valid selector because '<index>' is not a valid index</index></selector>
mg_s	elingorefyntaxitems which are actually present in the object	' <selector>' is not a valid selector because '<selector_segment>' is not a list.</selector_segment></selector>
	actually present in the object	' <selector>' is not a valid selector because '<selector_segment>' is not a</selector_segment></selector>
		property.
ob-	certain observable object	' <pre>'<pre>'<pre>can't resolve '<embed-< pre=""></embed-<></pre></pre></pre>
serv-		property>' reference ' <identifier>'</identifier>
	ologoare_atefepenoresobject	' <pre>'<pre>'<pre>reference 'datamets' '<pre>'<pre>'<pre>'</pre> must refer to an object of</pre></pre></pre></pre></pre>
		type ' <type(s)>'</type(s)>
ar-	the 'mime_type' property of	the 'mime_type' property of object ' <identifier>' ('<mime_type>') must be</mime_type></identifier>
ti-	artifact objects comes from	an IANA registered MIME Type of the form 'type/subtype'.
fact_i	mithree_Tieympeplate column in the	
_	IANA media type registry	
char-	certain properties of cy-	The 'path_enc' property of object ' <identifier>' ('<path_enc>') must be an</path_enc></identifier>
ac-	ber observable objects come	IANA registered character set.
ter_se	t from the IANA Character	The 'name_enc' property of object ' <identifier>' ('<name_enc>') must be</name_enc></identifier>
	Set list.	IANA registered character set.
lan-	the 'lang' property of SDOs	' <lang>' is not a valid RFC 5646 language code.</lang>
guage	is a valid RFC 5646 lan-	
	guage code	
soft-	the 'language' property of	The 'languages' property of object ' <identifier>' contains an invalid code</identifier>
ware_	languvague objects is a valid	(' <lang>').</lang>
	ISO 639-2 language code	
pat-	that the syntax of the pattern	' <object>' is not a valid observable type name</object>
terns	of an indicator is valid, and	Custom Observable Object type ' <object>' should start with 'x-' followed</object>
	that objects and properties	by a source unique identifier (like a domain name with dots replaced by
	referenced by the pattern	hyphens), a hyphen and then the name
	are valid. This runs the	Custom Observable Object type ' <object>' should start with 'x-'</object>
4.1. N	latidatory Checks - STIX 2.	1 ' <pre></pre>
	//github.com/oasis-open/cti-	Cyber Observable Object custom property ' <pre></pre>
	pattern-validator) to check	'x_' followed by a source unique identifier (like a domain name with dots
	the syntax of the pattern.	replaced by underscores), an underscore and then the name

4.2 Optional Checks - STIX 2.1

Code	Name	Ensures	Errors/Warnings
1	format-checks	all 1xx checks are run.	
		Specifically:	

Table 1 – continued from previous page

custom-prefix names of custom object Note: This checks func-

types, properties, observable object properties, and observable object properties, and observable object settlessions follow the correct format of the correct format object type should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by a source unique identifier (like a domain name with dots replaced by a source unique identifier (like a domain name with dots replaced by a source unique identifier (like a domain name with dots replaced by a source unique identifier (like a domain name with dots replaced by a source unique identifier (like a domain name with dots replaced by a source unique identifier (like a domain name with dots replaced by a source unique identifier (like a domain name with dots replaced by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type 'cobservable object-extensions' should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object extension type 'cobservable object oxidension property 'cobservable object oxidension property 'cobservable object oxidension property 'cobservable object oxidension type 'cobservable object oxidension property 'cobservable object oxidension and then the name. Cyber Observable Object oxidension should start with 'x.' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object oxidension should start with 'x.' followed by a source unique identifier (like a domain name with dots replaced by hyphens).	101	custom-prefix	names of custom object	Note: This checks func-
object properties, and observable object extensions follow the correct format disabled. custom object type 'cobjects' should start with 'x- followed by a source unique identifier (like a domain anne with dots replaced by a sphen, a hyphen and then the name. Custom property 'cpropertys' should have a type that starts with 'x-' followed by a source unique identifier (like a domain anne with dots replaced by a sphen), a hyphen and then the name. Custom object type 'cobservable object' should start with 'x-' followed by a source unique identifier (like a domain anne with dots replaced by a source unique identifier (like a domain anne with dots replaced by hyphens), a hyphen and then the name. Custom 'Cyber Observable object' extension' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable-object-extension's should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable object custom property 'cobservable object custom property 'cobservable object custom property 'costemplaced by hyphens), a hyphen and then the name. Cyber Observable object custom property 'costemplaced by hyphens), a hyphen and then the name. Cyber Observable object custom property 'costemplaced by hyphens), a hyphen and then the name. Cyber Observable object custom property 'cpropertys' should start with 'x-' followed by a source unique identifier (like a domain mame with dots replaced by hyphens), a hyphen and then the name. Cyber Observable object custom property 'cpropertys' in the extensions.			types, properties, observ-	tionality that has been
servable object extensions follow the correct format this check only runs if extensions—use (401) is disabled. custom objects' should start with 'x.' followed by a source unique identifier (file a domain name with dots replaced by hyphens), a hyphen and then the name. custom property 'cypropertys' should have a type that starts with 'x.' followed by a source unique identifier (file a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type 'cobservable_objects' should start with 'x.' followed by a source unique identifier (file a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Observable Object type 'cobservable-object-extensions' should start with 'x.' followed by a source unique identifier (file a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable-object-extensions' should start with 'x.' followed by a source unique identifier (file a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cobservable object-extensions' should start with 'x.' followed by a source unique identifier (file a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cobservable object-extensions' should start with 'x.' followed by a source unique identifier (file a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cpropertys' in the sextensions should start with 'x.' followed by a source unique identifier (file a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cpropertys' in the sextensions should start with 'x.' followed by a source unique identifier (file a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cpropertys' in the sextensions should start with 'x.' followed by a source unique (file of the file of the file			able objects, observable	deprecated and replaced
follow the correct format State Caston Caston Caston			object properties, and ob-	by extensions. Thus,
disabled. custom object type ' <object>'should start with 'x.' 'followed by a source unique identifier (fike a domain name with dots re- placed by hyphens), a hyphen and then the name. custom property '<pre>cryp>'s should have a type that starts with 'x.' fol- lowed by a source unique identifier (fike a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type 'cobserv- able object' should start with 'x.' followed by a source unique identifier (fike a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Oyber Observ- able Object extension type 'cobservable-object- extensions' should start with 'x.' followed by a source unique identifier (fike a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able Object custom property 'cobserv- able object crustom property 'cobserv- able object propertys' should start with 'x.' followed by a source unique identifier (fike a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able object propertys' should start with 'x.' followed by a source unique identifier (fike a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cobserv- able object propertys' should start with 'x.' followed by a source unique identifier (fike a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cprop- ertys' in the exettansions- extension should start3 with 'x.' followed by a source unique (fike a</pre></object>				
custom object type 'cob- ject' should start with 'x- 'followed by a source unique identifier (like a domain name with dots re- placed by hyphens), a hy- phen and then the name, custom property ' <pre>crys' should have a type that starts with 'x-' fol- lowed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type 'cobservable object' should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observ- able Object extension type 'cobservable-object- extensions' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able Object custom property '<observable' '<observ-="" (like="" a="" able="" and="" by="" clusted="" custom="" cyber="" domain="" dots="" hyphen="" hyphens),="" identifier="" name="" name.="" object="" object_property="" observ-="" property="" replaced="" source="" the="" then="" unique="" with="">' should start with 'x.' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able object_property>' should start with 'x.' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able object_property>' should start with 'x.' followed by a source unique identifier (like a domain on the extensions). Cyber Observable Object custom property 'cprope- crys' in the extensions. cxtension should start3 with 'x.' followed by a source unique (like a</observable'></pre>			follow the correct format	extensions-use (401) is
ject>'should start with 'x-' followed by a source unique identifier (like a domain name with dots re- placed by hyphens), a hy- phen and then the name. custom property 'cprop- erty>'s should have a type that starts with 'x_' followed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type 'sobserv- able_object>'s should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observ- able Object extension type 'sobservable-object- extensions' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able Object custom property 'sobserv- able_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber ObservableObject custom property 'sobserv- able_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cproperty>' 'srbope- erty>'' in the cetensions- extension should staf3 with 'x_' followed by a source unique (like a				disabled.
followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. custom property 'cpropertys' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type 'cobservable_objects' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type 'cobservable-object-extensions' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable-object-extensions' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-object-extensions' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cobservable Object out on property 'c				custom object type ' <ob-< th=""></ob-<>
unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. custom property 'cproperty's should have a type that starts with 'x_' followed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type 'cobservable_object's observable_object's observable_object's observable object extension type 'cobservable object extension's should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable object extension type 'cobservable-object's extension's should start with 'x_' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable object custom property 'cobservable object custom property 'copservable object custom property 'copservab				=
domain name with dots replaced by hyphens), a hyphen and then the name. custom property 'cproperty's' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable, objects' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Cyber Observable, objects extension type 'cobservable, object extension' should start with 'x_' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension' should start with 'x_' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cobservable object custom property 'copservaya' in the extensions. 4.2. Optional Checks - STIX 2.1				
placed by hyphens), a hyphen and then the name. custom property 'cpropertys' should have a type that starts with 'x.' fol- lowed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type 'cobserv- able objects' should start with 'x.' followed by a source unique identifier (like a domain then the name. Custom Cyber Observ- able Object extension type 'cobservable-object- extensions' should start with 'x.' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able Object custom property 'cobserv- able Object custom property 'sobserv- able object custom property 'sobserv- able object propertys' should start with 'x.' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able object propertys' should start with 'x.' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cpropertys' in the case of the property 'cpropertys' in the case of the property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name. Cyber Observable Object custom property 'cpropertys' in the case of the name.				÷ .
phen and then the name. custom property ' <pre>sproperty>'should have a type that starts with 'x_' fol- lowed by source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type '<observ- '<observable-object-="" '<pre="" 'x'="" 'x_'="" (like="" a="" able="" able_objects'="" and="" by="" custom="" cyber="" domain="" dots="" extension="" extensions'="" followed="" hyphen="" hyphens),="" identifier="" name="" name.="" object="" observ-="" property="" propertys'="" replaced="" should="" source="" start="" the="" then="" type="" unique="" with="">should start with 'x' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>representation of the property '</pre> **Cyber Observable Object custom property '<pre>representation of the property '</pre> **Cyber Observable Object custom property '<pre>representation of the property of</pre></observ-></pre>				
custom property 'cpropertys' should have a type that starts with 'x_' followed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object 'should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type 'cobservable-object-extensions' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-object-extensions' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-object_propertys' should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-object_propertys' is not start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property in the <extension 'x-'="" (like="" a="" and="" by="" domain="" dots="" followed="" hyphen="" hyphens),="" identifier="" name="" name.<="" replaced="" should="" source="" start="" th="" the="" then="" unique="" with=""><th></th><th></th><th></th><th></th></extension>				
ertys' should have a type that starts with 'x_' followed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type <observable_objects' '<observable-object-extension'="" 'cpropertys'="" 'x-'="" (like="" <creatension="" a="" and="" by="" custom="" cyber="" domain="" dots="" followed="" hyphen="" hyphens),="" identifier="" in="" mane="" name="" name.="" name.<="" object="" object_propertys'="" observable="" observable-object-customs="" observable-object-extension="" observable-object-extension's="" observable-object_propertys'="" property="" replaced="" should="" source="" start="" th="" the="" then="" type="" unique="" with=""><th></th><th></th><th></th><th>-</th></observable_objects'>				-
that starts with 'x_' followed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object by cobservable_object> should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable_object_extensions' should start with 'x_' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable_object_extensions' should start with 'x_' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain mane with dots replaced by hyphens), a hyphen and then the name. Cyber Observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property in the exetensions—the name. Cyber Observable Object custom property in the exetensions in the exetensions—the name. Cyber Observable Object custom property in the exetensions in the exetensions with 'x_' followed by a source unique (like a a domain name with dots replaced by hyphens), a hyphen and then the name.				
lowed by a source unique identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type 'cobservable_objects' should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable-object-extensions' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable-object-extensions' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable_object_custom property 'cobservable_object_propertys' should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable_object_custom property 'cytop-ertys' in the cextensions' a hyphen and then the name. Cyber Observable Object custom property 'cytop-ertys' in the cextensions' should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cytop-ertys' in the cextensions' start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cytop-ertys' in the cextensions' start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name.				
identifier (like a domain name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type ' <observable_object>' should start with 'x.' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable_object-extensions' should start with 'x.' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable_object-extensions' should start with 'x.' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable_object_custom property 'cobservable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable_object_property>' in the <extension 'x_'="" (like="" a="" by="" cyber="" domain="" followed="" identifier="" observable_object_property="" same.="" should="" source="" start="" unique="" with="">' cyproperty>' in the <extension 'x_'="" (like="" a="" by="" domain="" followed="" identifier="" same.<="" should="" source="" start="" th="" unique="" with=""><th></th><th></th><th></th><th></th></extension></extension></observable_object>				
name with dots replaced by a hyphen), a hyphen and then the name. Custom Observable Object type 'cobservable object's should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type 'cobservable-object-extensions' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-object-extensions able Object custom property 'cobservable-object_propertys' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-object_propertys' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-Object custom property 'cytop-ertys' in the <extension 'cytop-ertys'="" 'x_'="" (like="" (like<="" 3="" <extensions="" a="" by="" domain="" followed="" identifier="" in="" property="" should="" source="" starf="" th="" the="" unique="" with=""><th></th><th></th><th></th><th></th></extension>				
by a hyphen), a hyphen and then the name. Custom Observable Object type ' <observable_object' '<observable-object-extension="" 'x-'="" (like="" a="" and="" by="" custom="" cyber="" domain="" dots="" extension="" followed="" hyphen="" hyphens),="" identifier="" name="" name.="" object="" observable="" replaced="" should="" source="" start="" the="" then="" type="" unique="" with="">' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-object-extension>' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>reproperty>' in the <extension> a kyphen should staff with 'x_' followed by a source unique (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom should staff with 'x_' followed by a source unique (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom should staff with 'x_' followed by a source unique (like a domain name with dots replaced by hyphens), a hyphen and then the name.</extension></pre></pre></observable_object_property></observable_object'>				`
and then the name. Custom Observable Object type ' <observ- able_object="">' should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observ- able Object extension type '<observable-object- extension="">' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able Object custom property '<observ- able_object_property="">' should start with 'x.' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observ- able_object_property>' should start with 'x.' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>cypo- erty>' in the <extension> extension should start' with 'x.' followed by a source unique (like a</extension></pre></observ-></observable-object-></observ->				_
Custom Observable Object type ' <observable_object' '<observable="" '<observable_object_custom="" '<observable_object_propertyo'="" '<ol="" 'x-'="" (like="" <extension="" a="" and="" by="" custom="" cyber="" domain="" dots="" extension'="" followed="" hyphen="" hyphens),="" identifier="" in="" name="" name.="" object="" observable="" property="" propertyo'="" replaced="" should="" source="" start="" the="" then="" unique="" with=""> cyber Observable Object custom property ' cy</observable_object'>				
Object type ' <observable_object>' should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable-object-extension>' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-object-extension> a hyphen and then the name. Cyber Observable-object custom property '<observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-Object_custom property '<observable_object_property>' in the <extension> extension should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'y-property>' in the <extension> extension should startant 'x_' followed by a source unique (like a domain name with dots replaced by hyphens), a hyphen and then the name.</extension></extension></observable_object_property></observable_object_property></observable_object>				
able_object>' should start with 'x-' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type ' <observable-object-extension>' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<observable '<observable_object_property="" custom="" object="" property="">' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<property>' in the <extension> extension should start3 with 'x_' followed by a source unique (like a domain name with 'x_' followed by a source unique (like a domain name with dots replaced by hyphens), a hyphen and then the name.</extension></property></observable></observable-object-extension>				
with 'x.' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type 'cobservable-object-extensions' should start with 'x.' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cobservable Object custom property 'cobservable_object_propertys' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'copperitys' in the <extension 'cypropertys'="" 'x_'="" (like="" <extension="" a="" and="" by="" custom="" cyber="" domain="" dots="" followed="" hyphen="" hyphens),="" in="" name="" name.="" name.<="" object="" observable="" property="" replaced="" should="" source="" start="" th="" the="" then="" unique="" with=""><th></th><th></th><th></th><th></th></extension>				
source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type ' <observable-object-extensions' '<observable="" 'cypropertys'="" 'x-'="" 'x_'="" 'xobservable="" (like="" <extension="" <extensions="" a="" and="" by="" custom="" cyber="" domain="" dots="" followed="" hyphen="" hyphens),="" identifier="" in="" name="" name.="" name.<="" object="" object_custom="" object_propertys'="" observable="" of="" property="" replaced="" should="" source="" start="" th="" the="" then="" unique="" with=""><th></th><th></th><th></th><th></th></observable-object-extensions'>				
(like a domain name with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type ' <observable-object-extension's '<observable_object_property="" 'x.'="" (like="" a="" and="" by="" custom="" cyber="" domain="" dots="" followed="" hyphen="" hyphens),="" identifier="" name.="" object="" observable="" property="" replaced="" should="" source="" start="" the="" then="" unique="" with="">' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre> cyber Observable Object custom property '</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></observable_object_property></observable-object-extension's>				
with dots replaced by hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type ' <observable-object-extension's '<observable="" 'x-'="" (like="" a="" and="" by="" custom="" cyber="" domain="" dots="" followed="" hyphen="" hyphens),="" identifier="" name.="" object="" observable="" property="" replaced="" should="" source="" start="" the="" then="" unique="" with="">' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<rp> 4.2. Optional Checks - STIX 2.1 with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre> cyper Observable Object custom property '</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></rp></observable-object-extension's>				_
hyphens), a hyphen and then the name. Custom Cyber Observable Object extension type ' <observable-object-extension>' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<observable_object_property>' in the domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>crtys' in the <extension> cyber Observable Object custom property '<pre>crtys' in the <extension> domain should start 3 with 'x_' followed by a source unique (like a</extension></pre></extension></pre></observable_object_property></observable_object_property></observable-object-extension>				
then the name. Custom Cyber Observable Object extension type 'cobservable-object-extension>' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable-object custom property 'cobservable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'cobservable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property 'coproperty>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension>				
Custom Cyber Observable Object extension type ' <observable-object-extension>' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre> 'cyber Observable Object custom property '<pre> 'cyber Observable Object custom property '<pre> 'cyber Observable Object custom property '<pre> 'cytopservable Object custom p</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></observable_object_property></observable-object-extension>				
able Object extension type ' <observable-object-extensions' '<observable_object_property="" 'x-'="" (like="" a="" and="" by="" custom="" cyber="" domain="" dots="" followed="" hyphen="" hyphens),="" identifier="" name.="" object="" observable="" property="" replaced="" should="" source="" start="" the="" then="" unique="" with="">' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<observable_object_property>' and the source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>cyperty>' in the <extensions '<="" custom="" cyber="" extension="" object="" observable="" of="" pre="" property="" the=""> 4.2. Optional Checks - STIX 2.1</extensions></pre></observable_object_property></observable-object-extensions'>				
type ' <observable-object-extensions' '<observable_object_property="" 'x-'="" (like="" a="" and="" by="" custom="" cyber="" domain="" dots="" followed="" hyphen="" hyphens),="" identifier="" name.="" object="" observable="" property="" replaced="" should="" source="" start="" the="" then="" unique="" with="">' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre></pre></observable-object-extensions'>				
extension>' should start with 'x-' followed by a source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property ' <observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>cyber Observable Object custom property '<pre>crty>' in the <extension> extension should start33 with 'x_' followed by a source unique (like a</extension></pre></pre></observable_object_property>				
source unique identifier (like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property ' <observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>custom property '</pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></pre></observable_object_property>				
(like a domain with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property ' <observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>Cyber Observable Object custom property '<pre>rety>' in the <extension> extension should start3 with 'x_' followed by a source unique (like a</extension></pre></pre></observable_object_property>				with 'x-' followed by a
replaced by hyphens), a hyphen and then the name. Cyber Observ- able Object custom property ' <observ- able_object_property="">' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<property>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></property></observ->				source unique identifier
a hyphen and then the name. Cyber Observable Object custom property ' <observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre></pre></observable_object_property>				(like a domain with dots
name. Cyber Observable Object custom property ' <observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre> cyper Observable Object custom property '<pre> erty>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></pre></pre></observable_object_property>				
Cyber Observable Object custom property ' <observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<pre>crty>' in the <extension> extension should start3 with 'x_' followed by a source unique (like a</extension></pre></observable_object_property>				
able Object custom property ' <observ- able_object_property="">' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<property>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></property></observ->				
property ' <observable_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property '<property>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></property></observable_object_property>				•
able_object_property>' should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property ' <property>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></property>				
should start with 'x_' followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property ' <property>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></property>				
followed by a source unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property ' <pre>rety>' in the <extension> erty>' in the <extension 'x_'="" (like="" a="" a<="" by="" followed="" should="" source="" star13="" th="" unique="" with=""><th></th><th></th><th></th><th></th></extension></extension></pre>				
unique identifier (like a domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property ' <pre>crty>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></pre>				
domain name with dots replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property ' <property>' in the <extension> erty>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></extension></property>				
replaced by hyphens), a hyphen and then the name. Cyber Observable Object custom property ' <property>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></property>				
a hyphen and then the name. Cyber Observable Object custom property ' <pre>crty>' in the <extension> extension should start3 with 'x_' followed by a source unique (like a</extension></pre>				
name. Cyber Observable Object custom property ' <property>' in the <extension> 4.2. Optional Checks - STIX 2.1 extension should star13 with 'x_' followed by a source unique (like a</extension></property>				
Cyber Observable Object custom property ' <pre> custom property '<pre> erty>' in the <extension> extension should star13 with 'x_' followed by a source unique (like a</extension></pre></pre>				
custom property ' <pre>crty>' in the <extension> 4.2. Optional Checks - STIX 2.1 extension should start3 with 'x_' followed by a source unique (like a)</extension></pre>				
4.2. Optional Checks - STIX 2.1 erty>' in the <extension> extension should starf13 with 'x_' followed by a source unique (like a</extension>				
4.2. Optional Checks - STIX 2.1 extension should start3 with 'x_' followed by a source unique (like a				
with 'x_' followed by a source unique (like a	4.2. Optional Checks - S	TIX 2.1		-
a source unique (like a				
				domain name with dots

Table 1 – continued from previous page

102	custom-prefix-lax	same as 101 but more le-	Note: This checks func-
102	custom-prenx-rax		
		nient; no source identifier	tionality that has been
		needed in prefix	deprecated and replaced
			by extensions. Thus,
			this check only runs if
			extensions-use (401) is
			disabled.
			custom object type ' <ob-< td=""></ob-<>
			ject>' should start with 'x-
			'in order to be compatible
			with future versions of the
			STIX 2 specification.
			custom property ' <prop-< td=""></prop-<>
			erty>' should have a type
			that starts with 'x_' in
			order to be compatible
			with future versions of the
			STIX 2 specification.
			-
			Custom Observable
			Object type ' <observ-< td=""></observ-<>
			able_object>' should start
			with 'x-'.
			Custom Observable
			Object extension
			type ' <observable-< td=""></observable-<>
			object_extension>'
			should start with 'x-'.
			Cyber Observable Object
			custom property ' <prop-< td=""></prop-<>
			erty>' should start with
			'x_'.
			Cyber Observable Object
			custom property ' <em-< td=""></em-<>
			bedded_property>' in the
			<pre><pre><pre><pre>of the <ob-< pre=""></ob-<></pre></pre></pre></pre>
			ject> object should start
			with 'x_'.
			Cyber Observable Object
			custom property ' <prop-< td=""></prop-<>
			erty>' in the <extension></extension>
			extension should start
			with 'x_'.
			•
			ject custom property
			' <pre>'<pre>coperty>' in the</pre></pre>
			<extension_property></extension_property>
			property of the <exten-< td=""></exten-<>
			sion> extension should
			start with 'x_'.
<u> </u>			Continued on next page

Table 1 – continued from previous page

mended versions of UUID (v5 for SCOs, v4 for the rest) value <identifier> is not a valid UUIDv5 ID. Open-vocab-format values of open vocabularies follow the correct format kill-chain-names kill-chain-phase name and phase follow the correct format should be all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name 'sphase_name' should be all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name 'schain_name' should be all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name 'schain_name' should be all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name 'schain_name' should be all lowercase and use hyphens instead of spaces or underscores as word separators. it deploys the correct format spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the correct format spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or underscores as word separators. it deploys the spaces or under</identifier>	103	uuid-check	objects use the recom-	Cyber Observable ID
Company Comp	103	uulu-clieck		
rest) Given ID value <identifier> is not a valid UUIDv4 ID. Open-vocab-format values of open vocabularies follow the correct format values of open vocabularies follow the correct format values of spaces or underscores as word separators. Sill_chain_name value value</identifier>				
tifier> is not a valid UUIDv4 ID. open-vocab-format values of open vocabularies follow the correct format should be all lowercase and use hyphens instead of spaces or underscores as word separators. kill-chain-names kill-chain-phase name and phase follow the correct format should be all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name ' <phase_name' '<key_value="" '<phase_name'="" all="" and="" as="" be="" dictionary="" hyphens="" instead="" key,="" lowercase="" lowercase.="" of="" or="" phase_name="" separators.="" should="" spaces="" underscores="" use="" word="">' is not a good key value. Observable objects follow the correct format lowercase. phase_name '<phase_name' '<phase_name'="" all="" be="" lowercase.="" lowercase.<="" phase_name="" should="" td=""><td></td><td></td><td>· ·</td><td></td></phase_name'></phase_name'>			· ·	
open-vocab-format open-vocab-format open-vocab-format values of open vocabularries follow the correct format kill-chain-names kill-chain-phase name and phase follow the correct format kill-chain-phase name and phase follow the correct format kill-chain-names kill-chain-phase name and phase follow the correct format kill-chain-names kill-chain-phase name and phase follow the correct format kill-chain-names kill-chain-names kill-chain-phase name and phase follow the correct format kill-chain-name ' <chain_name>' should be all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name '<phase_name '<phase_name'="" 'sphase_nam<="" 'sphase_name'="" td=""><td></td><td></td><td>lest)</td><td></td></phase_name></chain_name>			lest)	
open-vocab-format values of open vocabularies follow the correct format values of spaces or underscores as word separators. kill-chain-names kill-chain-phase name and phase follow the correct format value value value inderscores as word separators. kill-chain-names kill-chain-phase name and phase follow the correct format value value value value value instead of spaces or underscores as word separators. phase_name value object separators. phase_name value value all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name value all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name value all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name value value all lowercase and use hyphens instead of spaces or underscores as word separators. value value value value all lowercase or underscores as word separators. value value value value value value value value value of spaces or underscores as word separators. value v				
ies follow the correct format Sill-chain-names kill-chain-phase name and phase follow the correct format chain-name kill-chain-name kill-chain-name chain-name	111		.1	
121	111	open-vocab-format		
121				
121			mat	•
Rill-chain-names Kill-chain-phase name and phase follow the correct format Kill_chain_name Kil				
kill-chain-names kill-chain-phase name and phase follow the correct format should be all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name ' <phase_name' '<phase_name'="" 'spha<="" 'sphase_name'="" td=""><td></td><td></td><td></td><td>*</td></phase_name'>				*
phase follow the correct format phase follow the correct format seal lowercase and use hyphens instead of spaces or underscores as word separators. phase_name ' <phase_name' '<phase<="" '<phase_name'="" td=""><td></td><td></td><td></td><td></td></phase_name'>				
be all lowercase and use hyphens instead of spaces or underscores as word separators. phase_name ' <phase_name' '<identifier="" 141="" 142="" 143="" all="" analysis="" and="" as="" be="" correct="" follow="" for="" format="" hyphens="" instead="" integers="" keys="" keys.="" lowercase="" lowercase.="" malware="" malware-analysis-product="" names="" non-negative="" object="" objects="" observable="" observable-dictionary-keys="" observable-object-keys="" of="" or="" product="" separators.="" servable="" should="" spaces="" the="" their="" underscores="" use="" word="">' should be all lowercase with words separated by dash. 149 windows-process-priority' follows the of object '<identifier>' follows the or follows</identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></phase_name'>	121	kill-chain-names	_	
hyphens instead of spaces or underscores as word separators. phase_name ' <phase_name>' should be all lowercase and use hyphens instead of spaces or underscores as word separators 141</phase_name>			1 *	
or underscores as word separators. phase_name ' <phase_name' '<p="" '<phase_name'="">as word separators ' 'able Object keys 'able Objects should use non-negative integers for their keys. ' 'able Objects should use non-negative integers for their keys. 'able Objects should use non-negative integers for their keys. ' 'able Objects should use non-negative integers for their keys. 'able Objects should use non-negative integers for their keys. 'able Objects should use non-negative integers for their keys. 'able Objects should use non-negative integers for thei</phase_name'>			format	be all lowercase and use
separators. phase_name ' <phase_name>' should be all lowercase and use hyphens instead of spaces or underscores as word separators observable-object-keys observable object keys follow the correct format observable-dictionary- keys dictionaries in cyber ob- servable objects follow the correct format observable-dictionary- keys dictionaries in cyber ob- servable objects follow the correct format malware-analysis-product malware analysis product names follow the correct format malware-analysis-product malware analysis product names follow the correct format windows-process- windows-process- windows-process- priority-format separators. phase_name '<phase- '<p="" '<phase-name="">a windows-protes - should be all coverage with words separated by dash. 149 windows-process- priority-format 'priority' follows the 'chex-y-value>' should be lowercase with words separated by dash. The 'priority' property of object '<identifier>'</identifier></phase-></phase_name>				hyphens instead of spaces
phase_name ' <phase_name '<p="" '<phase_name'="">as word separators 'add object should use non-negative integers for their keys. 'as a dictionary key, ''key_value>' should be lowercase. 'and object 'cidentifier>' should be all lowercase with words separated by dash. '149 windows-process-ext's priority' follows the of object 'cidentifier>' yoriority' follows the of object 'cidentifier>' yoriority' follows the of object 'cidentifier>' yoriority' follows the object 'cidentifier>' yoriority' follows the object 'cidentifier>' yoriority' follows the object 'cidentifier' yoriority' follows the object 'cidentifier' yoriority' property of object 'cidentifier' yoriority' property of object</phase_name>				or underscores as word
141 Observable-object-keys Observable object keys follow the correct format Servable objects should use non- negative integers for their keys. 142 Observable-dictionary- keys Servable objects follow the correct format Servable objects should use non- negative integers for their keys. 143 Malware-analysis-product names follow the correct format The 'product' property of object 'cidentifiers' should be all lowercase with words separated by dash. 149 Windows-process- priority-format Windows-process- vignority' follows the correct of object 'cidentifiers' of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of their product of th				separators.
141 Observable-object-keys Observable object keys follow the correct format Servable objects should use non- negative integers for their keys. 142 Observable-dictionary- keys Servable objects follow the correct format Servable objects should use non- negative integers for their keys. 143 Malware-analysis-product names follow the correct format The 'product' property of object 'cidentifiers' should be all lowercase with words separated by dash. 149 Windows-process- priority-format Windows-process- vignority' follows the correct of object 'cidentifiers' of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of object 'cidentifiers' 149 Windows-process- priority' follows the correct of their product of th				phase_name
be all lowercase and use hyphens instead of spaces or underscores as word separators 141 observable-object-keys observable object keys follow the correct format good key value. Observable Objects should use non- negative integers for their keys. 142 observable-dictionary- keys servable objects follow the correct format malware-analysis-product names follow the correct format malware-analysis product names follow the correct format with words separated by dash. 143 windows-process- with words separated by dash. 149 windows-process- priority-format property of object ' <identifier>' for object '<identifier>' for object '</identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier>				
hyphens instead of spaces or underscores as word separators 141 observable-object-keys observable object keys follow the correct format objects should use non- negative integers for their keys. 142 observable-dictionary- dictionaries in cyber observable objects follow the correct format objects follow the correct format objects follow the correct format of object ' <key_value>' should be lowercase. 143 malware-analysis-product names follow the correct format of object '<identifier>' should be all lowercase with words separated by dash. 149 windows-process- priority-format observable objects follows the 'priority' follows the of object '<identifier>' for object '</identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></identifier></key_value>				-
observable-object-keys observable object keys follow the correct format observable-object-keys follow the correct format observable-object-keys follow the correct format observable-object-keys follow the correct format observable-dictionary- keys dictionaries in cyber observable objects follow the correct format observable-dictionary- keys dictionaries in cyber observable objects follow the correct format malware-analysis-product names follow the correct format malware-analysis-product names follow the correct format windows-process- priority-format or underscores as word separators ' <key_value>' is not a good key value. Observ- able Objects should use non- negative integers for their keys. As a dictionary key, '<key_value>' should be lowercase. The 'product' property of object '<identifier>' should be all lowercase with words separated by dash. 149 windows-process- priority-format windows-process-ext's 'priority' follows the 'ckey_value>' is not a good key value. Observ- able Objects should use non- negative integers for their keys. As a dictionary key, '<key_value>' should be lowercase. The 'product' property of object '<identifier>' format' The 'priority' property of object '<identifier>'</identifier></identifier></key_value></identifier></key_value></key_value>				
observable-object-keys observable object keys follow the correct format observable-object-keys follow the correct format observable-object-keys follow the correct format observable objects should use non- negative integers for their keys. As a dictionary key, ' <key_value>' should be lowercase. The 'product' property of object '<identifier>' should be all lowercase with words separated by dash. windows-process- priority-format windows-process-ext's 'priority' follows the object should use non- negative integers for their keys. As a dictionary key, '<key_value>' should be lowercase. The 'product' property of object '<identifier>' should be all lowercase with words separated by dash. The 'priority' property of object '<identifier>'</identifier></identifier></key_value></identifier></key_value>				* -
follow the correct format good key value. Observable Objects should use non- negative integers for their keys. 142 observable-dictionary- keys servable objects follow the correct format malware-analysis-product mames follow the correct format malware analysis product names follow the correct format should be all lowercase with words separated by dash. 149 windows-process- priority-format priority' follows the correct format should be all lowercase with words separated by dash.				separators
follow the correct format good key value. Observable Objects should use non- negative integers for their keys. 142 observable-dictionary- keys servable objects follow the correct format malware-analysis-product mames follow the correct format malware analysis product names follow the correct format should be all lowercase with words separated by dash. 149 windows-process- priority-format priority' follows the correct format should be all lowercase with words separated by dash.	141	observable-object-keys	observable object keys	' <key_value>' is not a</key_value>
non- negative integers for their keys. 142 observable-dictionary- keys servable objects follow the correct format malware-analysis-product names follow the correct format 143 malware-analysis-product names follow the correct format should be all lowercase with words separated by dash. 149 windows-process- priority-format 'priority' follows the of object ' <identifier>' 149 vindows-process- priority-format 'priority' follows the of object '<identifier>'</identifier></identifier>			follow the correct format	
dictionaries in cyber observable-dictionary-keys servable objects follow the correct format malware-analysis-product names follow the correct format minutes follow the correct format should be all lowercase with words separated by dash. windows-process-priority-format priority' follows the correct of object ' <identifier>' format format windows-process-ext's priority-format priority' follows the correct of object '<identifier>' format format windows-process-ext's priority-format format format format format windows-process-ext's priority-format format forma</identifier></identifier>				able Objects should use
dictionaries in cyber observable-dictionary-keys servable objects follow the correct format malware-analysis-product names follow the correct format minutes follow the correct format should be all lowercase with words separated by dash. windows-process-priority-format priority' follows the correct of object ' <identifier>' format format windows-process-ext's priority-format priority' follows the correct of object '<identifier>' format format windows-process-ext's priority-format format format format format windows-process-ext's priority-format format forma</identifier></identifier>				non- negative integers for
keys servable objects follow the correct format lowercase. 143 malware-analysis-product mames follow the correct format should be all lowercase with words separated by dash. 149 windows-process-priority-format 'priority' follows the of object ' <identifier>' format format should be all lowercase with words separated by dash. 149 windows-process-priority-format 'priority' follows the of object '<identifier>'</identifier></identifier>				
the correct format lowercase. malware-analysis-product malware analysis product names follow the correct format should be all lowercase with words separated by dash. windows-process-priority-format priority' follows the of object ' <identifier>' format format format windows-process-ext's priority-format format form</identifier>	142	observable-dictionary-	dictionaries in cyber ob-	As a dictionary key,
malware-analysis-product malware analysis product names follow the correct format should be all lowercase with words separated by dash. windows-process-priority-format 'priority' follows the of object ' <identifier>' The 'product' property of object '<identifier>' should be all lowercase with words separated by dash. The 'priority' property of object '<identifier>'</identifier></identifier></identifier>		keys	servable objects follow	' <key_value>' should be</key_value>
names follow the correct format of object ' <identifier>' should be all lowercase with words separated by dash. 149 windows-process- priority-format 'priority' follows the of object '<identifier>' should be all lowercase with words separated by dash. The 'priority' property of object '<identifier>'</identifier></identifier></identifier>			the correct format	lowercase.
format should be all lowercase with words separated by dash. 149 windows-process- priority-format 'priority' follows the of object ' <identifier>'</identifier>	143	malware-analysis-product	malware analysis product	The 'product' property
with words separated by dash. 149 windows-process- priority-format 'priority' follows the of object ' <identifier>'</identifier>			names follow the correct	of object ' <identifier>'</identifier>
dash. 149 windows-process- priority-format 'priority' follows the of object ' <identifier>'</identifier>			format	should be all lowercase
dash. 149 windows-process- priority-format 'priority' follows the of object ' <identifier>'</identifier>				
priority-format 'priority' follows the of object ' <identifier>'</identifier>				
priority-format 'priority' follows the of object ' <identifier>'</identifier>	149	windows-process-	windows-process-ext's	The 'priority' property
		priority-format		of object ' <identifier>'</identifier>
			correct format	should end in '_CLASS'.

Table 1 – continued from previous page

	Table I – continued		
150	hash-length	keys in 'hashes'-type	Object ' <identifier>'</identifier>
		properties are not too long	has a 'hashes' dictio-
			nary with a hash of type
			' <hash_type>', which is</hash_type>
			longer than 30 characters.
			Object ' <identifier>' has</identifier>
			an NTFS extension with
			an alternate data stream
			that has a 'hashes' dictio-
			nary with a hash of type
			' <hash_type>', which is</hash_type>
			longer than 30 characters.
			Object ' <identifier>' has</identifier>
			a Windows PE Binary
			File extension with a file
			header hash of ' <hash>',</hash>
			which is longer than 30
			characters.
			Object ' <identifier>' has a</identifier>
			Windows PE Binary File
			extension with an optional
			header that has a hash of
			' <hash>', which is longer</hash>
			than 30 characters.
			Object ' <identifier>' has</identifier>
			a Windows PE Binary
			File extension with a sec-
			tion that has a hash of
			' <hash>', which is longer</hash>
			than 30 characters.
			Object ' <identifier>'</identifier>
			hash a 'hashes' dictio-
			nary with a hash of type
			' <hash_type>', which is</hash_type>
			longer than 30 characters.
2	approved-values	all 2xx checks are run.	<u> </u>
		Specifically:	
201	marking-definition-type	marking definitions use a	Marking definition 'def-
		valid definition_type	inition_type' should
			be one of: <marking-< td=""></marking-<>
			definition-type>.
L	I.	1	0

Table 1 – continued from previous page

		led from previous page	1
202	relationship-types	relationships are among those defined in the specification	' <object>' is not a suggested relationship source object for the '<relationship.' '<relationship="" relationship.="">' is not a suggested relationship type for '<object>' objects. '<object>' is not a suggested relationship target object for '<object>' objects with the '<relationship.' relationship.<="" td=""></relationship.'></object></object></object></relationship.'></object>
203	duplicate-ids	objects in a bundle with duplicate IDs have different <i>modified</i> timestamps	Duplicate ID ' <identifier>' has identical 'modified' timestamp. If they are different versions of the same object, they should have different 'modified' properties,</identifier>
210	all-vocabs	all of the following open vocabulary checks are run	' <pre>'<pre>'<pre>contains a value n the <vocab_name>- ov vocabulary.</vocab_name></pre></pre></pre>
211	attack-motivation	certain property values are from the attack-motivation vocabulary	' <pre>'<pre>'<pre>contains a value not in the attack- motivation-ov vocabulary</pre></pre></pre>
212	attack-resource-level	certain property values are from the attack-resource-level vocabulary	' <pre>'<pre>contains a value not in the attack- resource-level-ov vocabu- lary</pre></pre>
213	identity-class	certain property values are from the identity-class vo- cabulary	' <pre>'<pre>'<pre>contains a value not in the identity- class-ov vocabulary</pre></pre></pre>
214	indicator-types	certain property values are from the indicator-types vocabulary	' <pre>'<pre>'<pre>contains a value not in the indicator- types-ov vocabulary</pre></pre></pre>
215	industry-sector	certain property values are from the industry-sector vocabulary	' <pre>'<pre>'<pre>contains a value not in the industry- sector-ov vocabulary</pre></pre></pre>
216	malware-types	certain property values are from the malware-types vocabulary	' <pre>'<pre>contains a value not in the malware- types-ov vocabulary</pre></pre>
218	report-types	certain property values are from the report-types vo- cabulary	' <pre>'<pre>contains a value not in the report- types-ov vocabulary</pre></pre>
219	threat-actor-types	certain property values are from the threat-actor- types vocabulary	' <pre>'<pre>contains a value not in the threat- actor-types-ov vocabulary</pre></pre>
			Cantinuad an navt name

Table 1 – continued from previous page

		nom previous page	
220	threat-actor-role	certain property values are	' <pre>'<pre>contains a</pre></pre>
		from the threat_actor_role	value not in the threat-
		vocabulary	actor-role-ov vocabulary
221	threat-actor-sophistication	certain property val-	' <pre>'<pre>contains a</pre></pre>
		ues are from the	value not in the threat-
		threat_actor_sophistication	actor-sophistication-ov
		vocabulary	vocabulary
222	tool-types	certain property values are	' <pre>'<pre>contains a</pre></pre>
		from the tool_types vo-	value not in the tool-
		cabulary	types-ov vocabulary
223	region	certain property values are	' <pre>'<pre>contains a</pre></pre>
		from the region vocabu-	value not in the region-ov
		lary	vocabulary
225	grouping-context	certain property values are	' <pre>'<pre>contains a</pre></pre>
		from the grouping-context	value not in the grouping-
		vocabulary	context-ov vocabulary
226	implementation-	certain property values are	' <property>' con-</property>
	languages	from the implementation-	tains a value not in
		languages vocabulary	the implementation-
			languages-ov vocabulary
227	infrastructure-types	certain property values are	' <pre>'<pre>contains</pre></pre>
		from the infrastructure-	a value not in the
		types vocabulary	infrastructure-types-ov
		Ţ	vocabulary
228	malware-capabilities	certain property values	' <pre>'<pre>contains a</pre></pre>
		are from the malware-	value not in the malware-
		capabilities vocabulary	capabilities-ov vocabulary
230	processor-architecture	certain property values	' <pre>'<pre>contains a</pre></pre>
		are from the processor-	value not in the processor-
		architecture vocabulary	architecture-ov vocabu-
		, and the second se	lary
231	malware-result	certain property values are	' <pre>'<pre>contains a</pre></pre>
		from the malware-result	value not in the malware-
		vocabulary	result-ov vocabulary
	I.	•	•

Table 1 – continued from previous page

241		contain manager values and	Object 'sidentifican'
241	hash-algo	certain property values are	Object ' <identifier>'</identifier>
		from the hash-algo vocab-	has a 'hashes' dictio-
		ulary	nary with a hash of type
			' <hash_type>', which</hash_type>
			is not a value in the
			hash-algorithm-ov vocab-
			ulary nor a custom value
			prepended with 'x_'.
			Object ' <identifier>'</identifier>
			has an NTFS extension
			with an alternate data
			stream that has a 'hashes'
			dictionary with a hash
			-
			of type ' <hash_type>',</hash_type>
			which is not a value in
			the hash- algorithm-ov
			vocabulary nor a custom
			value prepended with
			'X_'.
			Object ' <identifier>'</identifier>
			has a Windows PE Bi-
			nary File extension with
			a file header hash of
			' <hash_type>', which is</hash_type>
			not a value in the hash-
			algorithm- vocabulary nor
			a custom value prepended
			with 'x_'.
			Object ' <identifier>' has a</identifier>
			Windows PE Binary File
			extension with an optional
			header that has a hash
			of ' <hash_type>', which</hash_type>
			is not a value in the
			hash-algorithm-ov vocab-
			_
			ulary nor a custom value
			prepended with 'x_'.
			Object ' <identifier>' has</identifier>
			a Windows PE Binary
			File extension with a
			section that has a hash
			of ' <hash_type>', which</hash_type>
			is not a value in the
			hash-algorithm-ov vocab-
			ulary nor a custom value
			prepended with 'x_'.
			Continued on next page

Table 1 – continued from previous page

Table 1 – continued from previous page				
243	windows-pebinary-type	certain property values are from the windows- pebinary-type vocabulary	Object ' <identifier>' has a Windows PE Binary File extension with a 'pe_type' of '<pe_type>', which is not a value in the windows-pebinary-type-ov vocabulary.</pe_type></identifier>	
244	account-type	certain property values are from the account-type vo-cabulary	Object ' <identifier>'is a User Account Object with an 'account_type' of '<account_type>', which is not a value in the account-type-ov vocabulary.</account_type></identifier>	
245	indicator-pattern-types	certain property values are from the pattern-type vo- cabulary	' <pre>'<pre>'<pre>contains a value not in the pattern- type-ov vocabulary</pre></pre></pre>	
270	all-external-sources	all of the following exter- nal source checks are run		
271	mime-type	file.mime_type is a valid IANA MIME type	The 'mime_type' property of object ' <identifier>' ('<mime_type>') should be an IANA registered MIME Type of the form 'type/subtype'.</mime_type></identifier>	
272	protocols	certain property values are valid IANA Service and Protocol names	The 'protocols' property of object ' <identifier>' contains a value ('<protocol>') not in IANA Service Name and Transport Protocol Port Number Registry.</protocol></identifier>	
273	ipfix	certain property values are valid IANA IP Flow In- formation Export (IPFIX) Entities	The 'ipfix' property of object ' <identifier>' contains a key ('<ipfix>') not in IANA IP Flow Information Export (IPFIX) Entities Registry.</ipfix></identifier>	
274	http-request-headers	certain property values are valid HTTP request header names	The 'request_header' property of object ' <identifier>' contains an invalid HTTP header ('<http_request_header>').</http_request_header></identifier>	
275	socket-options	certain property values are valid socket options	The 'options' property of object ' <identifier>' contains a key ('<option>') that is not a valid socket option (SOIICMPICMP6IIPIPV6I MCASTITCPIRLMP)_*.</option></identifier>	

Table 1 – continued from previous page

		i irom previous page	
276	pdf-doc-info	certain property values are	The 'document_info_dict'
		valid PDF Document In-	property of object ' <iden-< td=""></iden-<>
		formation Dictionary keys	tifier>' contains a key
			(' <key>') that is not a</key>
			valid PDF Document In-
			formation Dictionary key.
277	countries	certain property values	Location 'country' should
		are valid ISO 3166-1	be a valid ISO 3166-1
		ALPHA-2 codes	ALPHA-2 Code.
301	network-traffic-ports	network-traffic objects	The Network Traffic ob-
		contain both src_port and	ject ' <identifier>' should</identifier>
		dst_port	contain both the 'src_port'
		ast_port	and 'dst_port' properties.
302	extref-hashes	external references	External reference ' <src>'</src>
302	extrei-masnes	SHOULD have hashes if	has a URL but no hash.
			has a URL but no hash.
202		they have a url	Dealer des
303	indicator-properties	Indicator objects have	Both the name and
		both name and description	description properties
		properties	SHOULD be present.
304	deprecated-properties	certain properties which	Included property ' <prop-< td=""></prop-<>
		have been deprecated are	erty>' is deprecated
		not being used	within the indicated spec
			version.
305	extension-description	Extension Definitions	The 'description' property
		have a description prop-	SHOULD be populated.
		erty	1 1
306	extension-properties	Ensure toplevel-property-	For extensions of the
		extensions include the	'toplevel- property-
		extension_properties	extension' type, the
		property	'extension_properties'
		Fire	property SHOULD
			include one or more
			property names.
401	extensions-use	custom objects, proper-	Custom object type ' <ob-< td=""></ob-<>
701	CATCHSIOHS-USC	ties, and observable ex-	
		l .	ject>' should be imple-
		tensions have been im-	mented using an extension
		plemented with Extension	with an 'extension_type'
		Definitions	of 'new-sdo'.
			Custom property ' <prop-< td=""></prop-<>
			erty>' should be 'imple-
			mented using an extension
			with an 'extension_type'
			of 'property- extension' or
			'toplevel-property- exten-
			sion'.
			Custom Cyber Observ-
			able Object extension
			type ' <extension>' should</extension>
			be implemented using
			an 'extension_type' of
			'property-extension'.
1		1	property-extension.

Contributing

We're thrilled that you're interested in contributing to the stix2-validator! 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!

5.1 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-validator.git
```

2. Install develoment-related dependencies and set up git submodules:

```
$ cd cti-stix-validator
$ pip install -r requirements.txt
$ git submodule update --init --recursive
```

3. Install pre-commit git hooks:

```
$ pre-commit install
```

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

5.2 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).

5.3 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.

To run the tests in your current Python environment, use the pytest command from the root project directory:

```
$ pytest
```

This should show all of the tests that ran, along with their status.

You can run a specific test file by passing it on the command line:

```
$ pytest stix2validator/test/test_<xxx>.py
```

To ensure that the test you wrote is running, you can deliberately add an assert False statement at the beginning of the test. This is another benefit of TDD, since you should be able to see the test failing (and ensure it's being run) before making it pass.

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=stix2validator
$ 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.

5.4 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 6

Indices and tables

- genindex
- modindex
- search