MISP-STIX

How to survive to STIX parsing?

MISP core team - Christian Studer
TLP:CLEAR

MISP Project
https://www.misp-project.org/

PASS THE SALT 2023

Who am I

- : chrisr3d
- : @chrisr3d@infosec.exchange
- : chrisred_68

- Interoperability Wizard @ CIRCL
- MISP core development team
- STIX SC co-chair

- & enthusiast

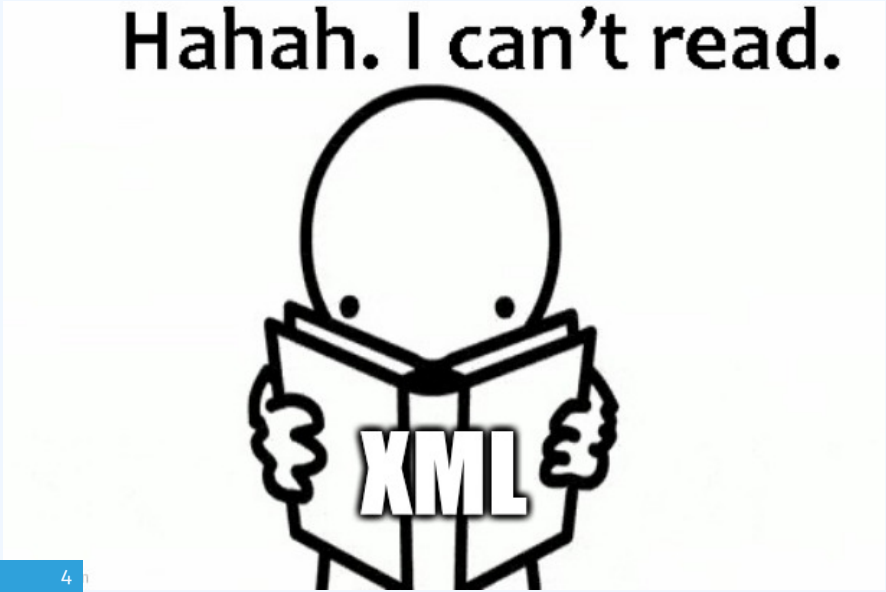
Summary

- A quick recap
- From an ocean of unknown errors...
  $\Rightarrow$ the difficulty to parse STIX content
- ... To a more & more accurate support
  $\Rightarrow$ *misp-stix* - The Holy Grail for MISP & STIX
- ... And even further
  $\Rightarrow$ Evolution & improvement perspectives
- The magic word: *interoperability*
- Examples

STIX - Quick recap

- **S**tructured **T**hreat **I**ntelligence E**x**pression
  - Focused on **Threat Intelligence** exchange
  - 2 major versions with different formats
    - 1.x - *mainly* XML
    - 2.x - *mostly* JSON
- **T**rusted **A**utomated E**x**change of **I**ntelligence **I**nformation
  - Exchange Protocol
  - Specifically designed to support the exchange of **CTI** represented in STIX

STIX 1.x - A TOUGH BEAST TO HANDLE



STIX 1.x - A TOUGH BEAST TO HANDLE

- Excessive complexity in certain advanced XML constructs
  - Difficult to implement & parse
- Multiple ways to represent information
  - Challenging for interoperability
- A plethora of different objects
  - Only a common subset of capabilities widely used
  - Many others poorly understood and in many cases never used
- A majority of properties are optional
  - Parsing challenges for consumers of STIX 1 content

STIX 2.x - an improvement

STIX 2.x - an improvement

- Lightweight & flattened representation of the objects
- More required properties
  - Easier to parse
- Extension definitions
  - More flexibility

STIX 2.x - the (still not perfect) improvement

- Lightweight & flattened representation of the objects
- More required properties
  - Easier to parse
- Extension definitions
  - More flexibility
- Number of objects reduced to a set of well-understood features
  - ⊕ Clearer for everyone
  - ⊖ Some definitions lost in the process
- Introduction of patterns within Indicator objects
  - ⊕ Ability to use different patterning languages (STIX 2.1)
  - ⊖ Observations and Indicators require alternate parsing implementations
- Still multiple ways to represent the same data

The reality about STIX parsing

Struggling with various STIX pattern creation designs

- Handling the multiple ways of reprensenting the *same* concept

```
"pattern": "[domain-name:resolves_to_refs.value = '8.8.8.8' AND \
            domain-name:value = 'google.com']"
"pattern": "[ipv4-addr:value = '8.8.8.8' AND \
            domain-name:value = 'google.com']"
"pattern": "[network-traffic:dst_ref.type = 'ipv4-addr' AND \
            network-traffic:dst_ref.value = '8.8.8.8' AND \
            domain-name:value = 'google.com']"
"pattern": "[network-traffic:dst_ref.type = 'domain-name' AND \
            network-traffic:dst_ref.value = 'google.com' AND \
            ipv4-addr:value = '8.8.8.8']"
```

- Understanding the meaning of data

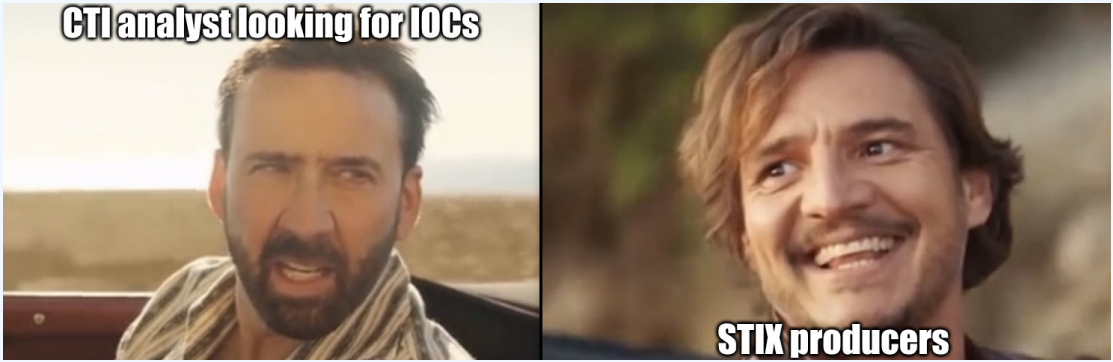
```
"pattern": "[mutex:defanged NOT = false]"
"pattern": "[network-traffic:src_packets NOT = 965283]"
"pattern": "([ipv4-addr:value = '3.162.166.34'] FOLLOWEDBY [network-traffic:is_active = false])"
```

Struggling with various STIX pattern creation designs

```python
from stix2.v21.bundle import Bundle
from stix2.v21.sdo import Indicator

patterns = (
    "[domain-name:resolves_to_refs.value = '8.8.8.8'
    "[ipv4-addr:value = '8.8.8.8' AND domain-name:va
    "[network-traffic:dst_ref.type = 'ipv4-addr' AND
    "[network-traffic:dst_ref.type = 'domain-name' A
    "[mutex:defanged NOT = false]",
    "[network-traffic:src_packets NOT = 965283]",
    "([ipv4-addr:value = '3.162.166.34'] FOLLOWEDBY
)


def populate_indicators(patterns):
    for n, pattern in enumerate(patterns):
        yield Indicator(
            name=f'indicator_{n}',
            description=f'Indicator #{n}',
            pattern=pattern,
            pattern_type='stix'
        )


bundle = Bundle(*populate_indicators(patterns))
with open('tmp/test_bundle.stix21.json', 'w') as f:
    f.write(bundle.serialize(indent=4))
```

```
oui chrisr3d ~/git/MISP/MISP-STIX-Converter
$ (git::dev) poetry run stix2_validator
tmp/test_bundle.stix21.json
============================================
====================================
[-] Results for: tmp/test_bundle.stix21.json
[+] STIX JSON: Valid
```

The unbearable weight of a massive mess



The constant validation issues

- We want to **keep UUIDs** for referencing
- Not everyone validates their content properly

The constant validation issues

- We want to **keep UUIDs** for referencing
- Not everyone validates their content properly
- Issues with UUIDs validation
  - Unable to load content

An easy fix - Making the UUIDs validation more flexible

- STIX 2 python library fork[1]
  - No change on the content validation
  - Differs only on the UUIDs validation

  $\Rightarrow$ Same UUID requirements on MISP & STIX

- Handling the "*worst*" UUIDs
  - Generating a v5 UUID to be used as the new identifier
  - Keeping a reference to the initial UUID

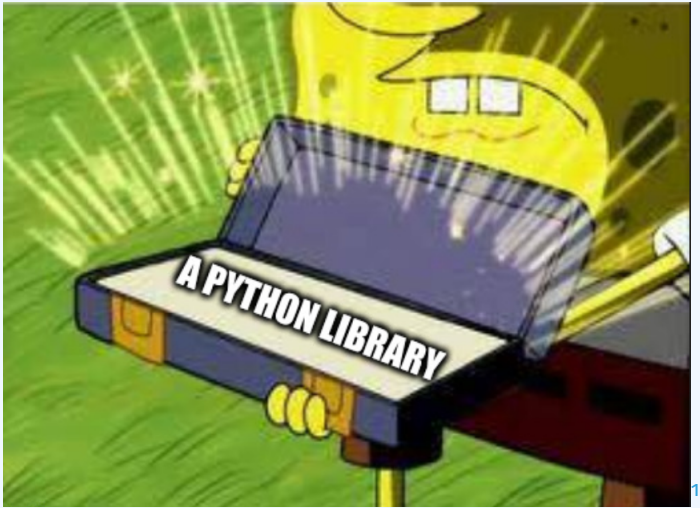
[1] https://github.com/MISP/cti-python-stix2 - https://pypi.org/project/misp-lib-stix2/

The infinite madness of empty references

- TAXII is designed to give STIX objects
- A STIX file can include a wide variety of information
- No check on the references
  - The TAXII server doesn't need to know
  - Neither does a STIX file
- MISP needs to get the information from the data we ingest


```
{
    type: threat-actor,
    spec_version: 2.1,
    id: threat-actor--8b6297fe-cae7-47c6-9256-5584b417849c,
    created_by_ref: identity--b38dfe21-7477-40d1-aa90-5c8671ce51c
    created: 2017-04-27T16:18:24.318Z,
    modified: 2017-04-27T16:18:24.318Z,
    name: The Joker,
    threat_actor_types: [
        terrorist,
        criminal
    ],
    aliases: [
        Joe Kerr,
        The Clown Prince of Crime
    ],
    roles: [
        director
    ],
    resource_level: team,
    primary_motivation: personal-satisfaction,
    object_marking_refs: [
        marking-definition--5e57c739-391a-4eb3-b6be-7d15ca92d5ed
    ]
},
```

misp-stix - The Holy Grail for MISP & STIX interactions



[1]

[1] https://github.com/MISP/misp-stix - https://pypi.org/project/misp-stix/

misp-stix - The Holy Grail for MISP & STIX interactions

- Used in MISP
  - Conversion only
- Can be used as a **stand-alone** tool [1]
  - Converting input file(s), saving results in output file(s)
- Enabling automation with python code
  - Handles both conversion and input(s)/output(s)
  - Supports all the available input formats
    - file names, JSON, PyMISP, STIX Packages or Bundles
- A complete mapping documentation[2]

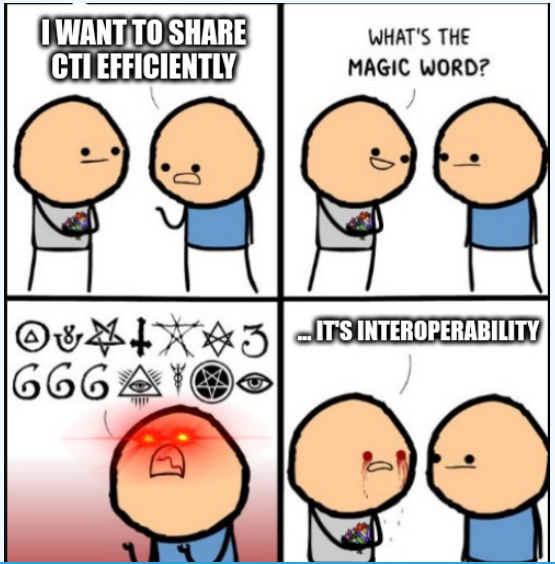
[1] i.e Command line

[2] https://github.com/MISP/misp-stix/tree/main/documentation

Continuous work

- Members of the OASIS CTI TC
  - Co-chairing
    - Supported changes to make the TC go for the open source way
  - Participating to the development process
    - Working Groups
  - More visibility & easier to contribute
- Collaboration with STIX users/producers
  - Get feedback & provide support
  - Fill the mapping gaps

The Magic Word



Examples - Command line help

```
oui chrisr3d ~/git/MISP/MISP-STIX-Converter
$ (git::dev) poetry run misp_stix_converter export -h
usage: misp_stix_converter export [-h] -f FILE [FILE ...] -v {1.1.1,1.2,2.0,2.1} [-s] [-m] [--output_dir OUTPUT_DIR] [-o OUTPUT_NAME] [--level {attribute,event}]
                                  [--format {json,xml}] [-n NAMESPACE] [-org ORG]

options:
  -h, --help            show this help message and exit
  -f FILE [FILE ...], --file FILE [FILE ...]
                        Path to the file(s) to convert.
  -v {1.1.1,1.2,2.0,2.1}, --version {1.1.1,1.2,2.0,2.1}
                        STIX specific version.
  -s, --single_output   Produce only one result file (in case of multiple input file).
  -m, --in_memory       Store result in memory (in case of multiple result files) instead of storing it in tmp files.
  --output_dir OUTPUT_DIR
                        Output path - used in the case of multiple input files when the `single_output` argument is not used.
  -o OUTPUT_NAME, --output_name OUTPUT_NAME
                        Output file name - used in the case of a single input file or when the `single_output` argument is used.

STIX 1 specific arguments:
  --level {attribute,event}
                        MISP data structure level.
  --format {json,xml}   STIX 1 format.
  -n NAMESPACE, --namespace NAMESPACE
                        Namespace to be used in the STIX 1 header.
  -org ORG              Organisation name to be used in the STIX 1 header.
oui chrisr3d ~/git/MISP/MISP-STIX-Converter
$ (git::dev) poetry run misp_stix_converter import -h
usage: misp_stix_converter import [-h] -f FILE [FILE ...] -v {1,2} [-s] [-o OUTPUT_NAME] [--output_dir OUTPUT_DIR] [-d DISTRIBUTION] [-sg SHARING_GROUP] [--galaxies_as_tags]

options:
  -h, --help            show this help message and exit
  -f FILE [FILE ...], --file FILE [FILE ...]
                        Path to the file(s) to convert.
  -v {1,2}, --version {1,2}
                        STIX major version.
  -s, --single_output   Produce only one MISP event per STIX file(in case of multiple Report, Grouping or Incident objects).
  -o OUTPUT_NAME, --output_name OUTPUT_NAME
                        Output file name - used in the case of a single input file or when the `single_output` argument is used.
  --output_dir OUTPUT_DIR
                        Output path - used in the case of multiple input files when the `single_output` argument is not used.
  -d DISTRIBUTION, --distribution DISTRIBUTION
                        Distribution level for the imported MIPS content.
  -sg SHARING_GROUP, --sharing_group SHARING_GROUP
                        Sharing group ID when distribution is 4.
  --galaxies_as_tags    Import MISP Galaxies as tag names instead of the standard Galaxy format.
oui chrisr3d ~/git/MISP/MISP-STIX-Converter
$ (git::dev)
```

Examples - Command line usage

- Conversion of STIX files

```
oui chrisr3d ~/git/MISP/MISP-STIX-Converter
$ (git::dev) poetry run misp_stix_converter import -v 2 -f tmp/debug/STIX/playbook_json/*.json
Failed parsing the following - and the related error message:
 - tmp/debug/STIX/playbook_json/automated-libra.json -  Invalid value for Indicator 'pattern': FAIL: Error found at line 1:0. input is missing square
 brackets
Successfully processed your files. Results available in:
 - tmp/debug/STIX/playbook_json/adept-libra.json.out
 - tmp/debug/STIX/playbook_json/agedlibra.json.out
 - tmp/debug/STIX/playbook_json/agent-tesla.json.out
 - tmp/debug/STIX/playbook_json/alloytaurus.json.out
 - tmp/debug/STIX/playbook_json/api-hammering-technique.json.out
 - tmp/debug/STIX/playbook_json/atlassian-confluence-CVE-2022-26134.json.out
 - tmp/debug/STIX/playbook_json/avoslocker-ransomware.json.out
 - tmp/debug/STIX/playbook_json/blackbasta-ransomware.json.out
 - tmp/debug/STIX/playbook_json/blackcat-ransomware.json.out
 - tmp/debug/STIX/playbook_json/bluesky-ransomware.json.out
 - tmp/debug/STIX/playbook_json/boggyserpens.json.out
 - tmp/debug/STIX/playbook_json/brute-ratel.json.out
 - tmp/debug/STIX/playbook_json/chromeloader.json.out
 - tmp/debug/STIX/playbook_json/clean-ursa.json.out
 - tmp/debug/STIX/playbook_json/cloaked-ursa.json.out
 - tmp/debug/STIX/playbook_json/clop-ransomware.json.out
 - tmp/debug/STIX/playbook_json/conti-ransomware.json.out
 - tmp/debug/STIX/playbook_json/crawling-taurus.json.out
 - tmp/debug/STIX/playbook_json/crooked-pisces.json.out
 - tmp/debug/STIX/playbook_json/darkside-ransomware.json.out
 - tmp/debug/STIX/playbook_json/dearcry-ransomware.json.out
 - tmp/debug/STIX/playbook_json/egregor-ransomware.json.out
 - tmp/debug/STIX/playbook_json/ekans-ransomware.json.out
 - tmp/debug/STIX/playbook_json/emotet.json.out
 - tmp/debug/STIX/playbook_json/evasive-serpens.json.out
 - tmp/debug/STIX/playbook_json/f5-big-ip-cve-2022-1388.json.out
 - tmp/debug/STIX/playbook_json/fighting-ursa.json.out
 - tmp/debug/STIX/playbook_json/golfing-taurus.json.out
 - tmp/debug/STIX/playbook_json/granite-taurus.json.out
 - tmp/debug/STIX/playbook_json/hellokitty-ransomware.json.out
 - tmp/debug/STIX/playbook_json/hermeticwiper.json.out
 - tmp/debug/STIX/playbook_json/hive-ransomware.json.out
```

- The MISP OSINT feed converted in STIX 2.1 format:
  `https://codeberg.org/adulau/misp-circl-feed`

Thank you for your attention

- How to report issues / ask questions?
  - **https://github.com/MISP/misp-stix/issues**
  - https://github.com/MISP/MISP/issues
- More information
  - **https://github.com/MISP/misp-stix/tree/main/documentation**
  - https://www.misp-project.org/blog/
- Follow updates on MISP
  - @: https://misp-community.org/@misp
  - : https://twitter.com/MISPProject