

STIX Patterning: Viva la revolución!

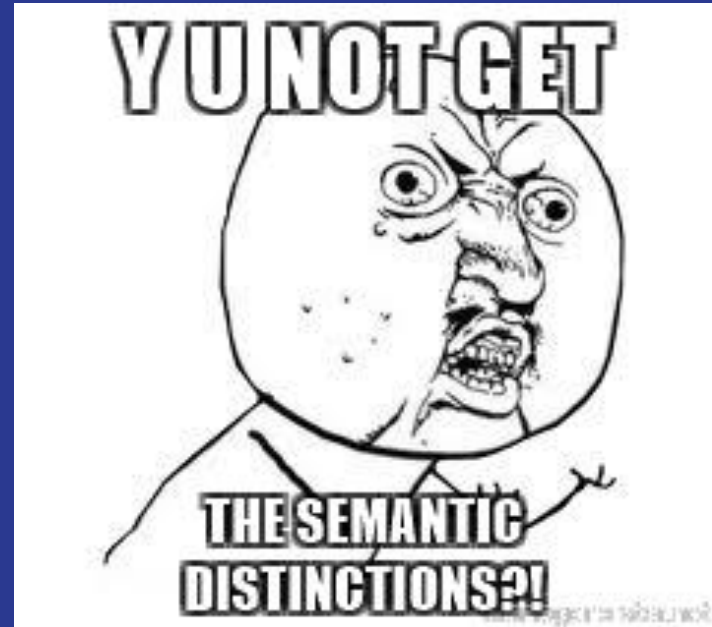
Cyber Threat Intelligence Matters
FIRST Technical Symposium and OASIS Borderless Cyber
Conference

Jason Keirstead - STSM, IBM Security

Trey Darley - Director of Standards Development, New Context

History vs. mystery

"What the hell were you guys smoking?!"



What was wrong in STIX 1.x

- Too many ways to define matches (multiple meanings of "Equals")
- Too many ways to define expressions (ANDs and ORs in *both* Indicators and Observables)
 - One analysis found twelve different ways to compare two IP addresses
- Lists are just plain "weird" (##comma## - need I say more?)
- Despite all this complexity, lacked fundamental capabilities such as temporal matching (A followed by B)

But (Snort|YARA|OpenIOC|Sigma) already exist?!

- Snort only makes sense on the network
- YARA library only works on a file-like blob
 - Neither allows encoding of malware behaviour information
- OpenIOC limited in expressivity; also limited in network coverage
- Basic use case: malware matching signature **X** will beacon with traffic that looks like **Y** before dropping **Z**
 - Combination of file attributes, network attributes, sequential / temporal matching
 - This extremely simple use case is **impossible** to model using any of these standards
- Sigma: <https://github.com/Neo23x0/sigma>
 - Their effort started after we'd already achieved our Committee Specification Draft. We reached out to collaborate but got zero acknowledgement. :-(


Questions we asked

- Should we think beyond simple CTI use cases of "find this IOC" ?
- What if our cybersecurity tools could share rules and searches for analytics and correlations?
- What factors have been preventing this from emerging in the industry? Could we have an opportunity to finally move the needle?
- What if SIEM vendor lock-in were to just die in a fire?

"We're here to put a dent in the universe." — Steve Jobs



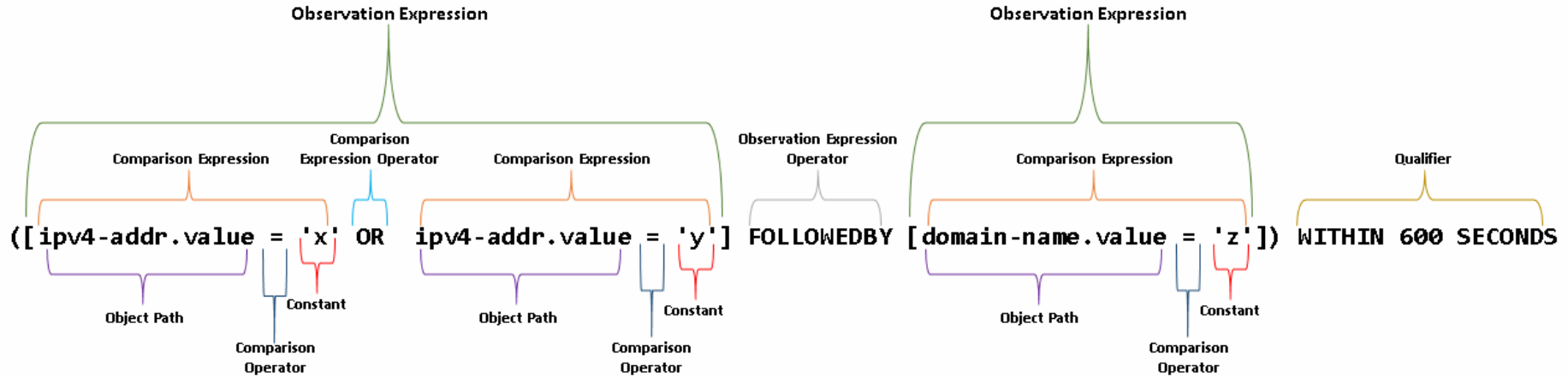
Basic design principles

- One way to do things (not 12)!
 - Base things on a **grammar**, not nested XML or JSON
 - Makes things easier for humans to understand, **and** for machines to parse!
 - Base that grammar on something that as many folks are familiar with as possible
 - Candidates: SQL, Lucene, YARA, Snort/OpenSig...
 - We ended with SQL-like after some debate
 - Define a grammar that allows sharing descriptions of advanced threats, not just simple atomic IOCs (ip = 1.2.3.4)
 - Define it in a way that was expandable in the future without "breaking changes"
- 

Overview of STIX Patterning

"What's the frequency, Kenneth?"


Basic structure of a STIX Pattern



How this ties to STIX Cyber Observables

- Cyber Observables provide a data model for describing things you've *actually* seen.
- STIX Patterning is a language for describing chaotic maliciousness one *might* see.
- SCO (STIX Cyber Observables) : nouns :: STIX Patterning : language
- SCO : DB Tables :: STIX Patterning : SQL



The background is a solid pink color. In the top right corner, there are several overlapping geometric shapes, including triangles and squares, in various shades of pink and magenta, creating a modern, abstract design.

THIS SOUNDS INCREDIBLY
COMPLICATED, I JUST
WANTED TO FIND AN IP
ADDRESS

It's not that bad, see!

Finding an IP

```
[ip-addr.value = '8.8.8.8']
```

Finding a URL

```
[url:value MATCHES  
'^(?:https?:\\/\\/)?(?:www\\.)?example\\.com\\/.*']
```

Finding one of two registry keys

```
[windows-registry-key:key =  
'HKEY_CURRENT_USER\\Software\\CryptoLocker\\Files'  
 OR windows-registry-key:key =  
'HKEY_CURRENT_USER\\Software\\Microsoft\\CurrentV  
ersion\\Run\\CryptoLocker_0388']
```

Currently-defined Cyber Observables

- Artifact
- AS
- Directory
- Email Address
- Email Message
- File
 - Archive Extension
 - NTFS File Extension
 - PDF File Extension
 - Raster Image File Extension
 - Windows PE Binary File Extension
- IPv4 Address
- IPv6 Address
- MAC Address
- Mutex
- Network Traffic
 - HTTP Request Extension
 - ICMP Extension
 - Network Socket Extension
 - TCP Extension
- Process
 - Windows Process Extension
 - Windows Service Extension
- Software
- User Account
 - UNIX Account Extension
- Windows Registry Key
- X.509 Certificate

Use cases and examples



File-based Pattern (vs. YARA)

Basic File with Hexadecimal Payload

STIX Indicator Pattern

```
[file:contents_ref.payload_bin MATCHES '\\x65\\x78\\x61\\x6d\\x70\\x6c\\x65' AND file:size > '31284']
```

Corresponding YARA Rule

```
rule Example
{
  strings:
    $hex_string = { 65 78 61 6d 70 6c 65 }

  condition:
    $hex_string and filesize > 31284
}
```

Basic File with Textual Payload


STIX Indicator Pattern

```
[file:contents_ref.payload_bin MATCHES 'this is an example']
```

Corresponding YARA Rule

```
rule Example
{
  strings:
    $text_string = "this is an example"

  condition:
    $text_string
}
```



Network-based Pattern (vs. Snort)

Basic TCP Network Traffic

STIX Indicator Pattern

```
[network-traffic:src_ref.type = 'ipv4-addr' AND network-traffic:src_ref.value = '192.0.2.1' AND network-traffic:dst_ref.type = 'ipv4-addr' AND network-traffic:dst_ref.value = '203.0.113.10' AND network-traffic:dst_port = '21' AND network-traffic:protocols[*] = 'tcp']
```

Corresponding Snort Rule

```
alert tcp 192.0.2.1 any -> 203.0.113.10 21
```

HTTP Network Traffic with User Agent

STIX Indicator Pattern

```
[network-traffic:dst_ref.type = 'ipv4-addr' AND network-traffic:dst_ref.value = '203.0.113.11' AND network-traffic:dst_port = '80' AND network-traffic:protocols[*] = 'tcp' AND network-traffic:extended_properties.http_ext.request_header.User-Agent = 'Mazilla/5.0']
```

Corresponding Snort Rule

```
alert tcp any any -> 203.0.113.11 80 (content:"User-Agent|3a|Mazilla/5.0"; http_header;)
```

Watching for "Fileless" UAC Bypass

```
[  
  ( windows-registry-key:key =  
    'HKEY_CURRENT_USER\\Software\\Classes\\exefile\\shell\\runas\\command' AND windows-registry-  
    key:values[*].name = 'isolatedCommand' )  
]
```

OR

```
[  
  ( windows-registry-key:key = 'HKEY_CURRENT_USER\\Microsoft\\Windows\\CurrentVersion\\App  
  Paths\\control.exe' AND windows-registry-key:values[*].data != "C:\\Windows\\System32\\cmd.exe" )  
]
```


Bad Powershell!

Suspicious Powershell has been used

[

```
process:command_line MATCHES  
'((.*NewObject(System)?NetWebClient.*DownloadFile.*((StartProcess)|(shellexecute)|(win32_process)|(start)|(saps)).*)|(.*)((iex)|(InvokeExpression)).*NewObject(System)?NetWebClient.*DownloadString.*)|(.NewObject(System)?NetWebClient.*DownloadString.*((iex)|(InvokeExpression)).*)|(.IEX.*\[SystemDiagnosticsProcess\\]\:\:Start.*)|(.StartBitsTransfer.*InvokeItem.*))'
```

]



Necurs Botnet

Looks for a particular malware payload followed by HTTP beaconing traffic generated by the payload:

```
[file:name = 'rekakva32.exe' AND file:parent_directory_ref.path MATCHES  
'C:\\Users\\[\\w\\s]+\\AppData\\Local\\Temp'] FOLLOWEDBY [network-  
traffic:protocols[*] = 'http' AND network-traffic:extensions.'http-request-  
ext'.request_method = 'post' AND network-traffic:extensions.'http-request-  
ext'.request_header.'User-Agent' = 'Windows-Update-Agent']
```

Source: <https://isc.sans.edu/forums/diary/Necurs+Botnet+malspam+pushes+Locky+using+DDE+attack/22946/>

OSS tools and libs

GITHUB



Github all the things!

OASIS Open Repository: TAXII 2 Server Library Written in Python

cti-taxii-server: <https://github.com/oasis-open/cti-taxii-server>

OASIS Open Repository: TAXII 2 Client Library Written in Python

cti-taxii-client: <https://github.com/oasis-open/cti-taxii-client>

OASIS Open Repository: Python APIs for STIX 2

cti-python-stix2: <https://github.com/oasis-open/cti-python-stix2>

OASIS Open Repository: Match STIX content against STIX patterns

cti-pattern-matcher: <https://github.com/oasis-open/cti-pattern-matcher>

OASIS Open Repository: Convert STIX 1.2 XML to STIX 2.0 JSON

cti-stix-elevator: <https://github.com/oasis-open/cti-stix-elevator>



Github all the things (2)!

Translate STIX 2 Patterning Queries Into Splunk and ElasticSearch

stix2patterns_translator: https://github.com/mitre/stix2patterns_translator

Downgrade STIX2 content to STIX1

cti-stix-slider: <https://github.com/oasis-open/cti-stix-slider>

Malware Information Sharing Platform & Threat Sharing

MISP: <https://github.com/MISP/MISP>

A cyber threat intelligence server based on TAXII 2 and written in Golang

freetaxii-server: <https://github.com/freetaxii/freetaxii-server>

APIs for generating STIX 2.x messages with Go (Golang)

libstix2: <https://github.com/freetaxii/libstix2>

The CaRT file format is used to store/transfer malware and its associated metadata

cse cart: <https://bitbucket.org/cse-assemblyline/cart>

Github all the things (3)!

Convert STIX2 to GraphML or GEXF (Gephi format)

StixConvert: <https://github.com/workingDog/StixConvert>

Convert STIX2 and load into Neo4j graph database

StixToNeoDB: <https://github.com/workingDog/StixToNeoDB>

Browser-based STIX2 editor, with ability to publish to a TAXII2 server

cyberstation: <https://github.com/workingDog/cyberstation>

STIX2 Scala library

scalastix: <https://github.com/workingDog/scalastix>

TAXII2 Scala library

Taxii2LibScala: <https://github.com/workingDog/Taxii2LibScala>

TAXII2 JS library


taxii2lib: <https://github.com/workingDog/taxii2lib>





We're Not Done!

Beyond indicators - analytics use cases

- Threat Intelligence sharing has received a lot of focus; however the analytics to actually **find** things, not so much
 - People re-build the same analytics over and over because they either don't know of, or have access to, what has been done many times before
 - **In order to share analytics in a scalable fashion, a vendor-neutral language for said analytics has to be developed**
 - **We believe SCO Pattern could be the basis for this**
 - **CAR** - The MITRE Cyber Analytics Repository
 - PRE-ATT&CK and ATT&CK based analytics
 - Long-term goal: ability to define the analytics in STIX Patterning
 - Collaborative ecosystem for analytics development
- 

Correlation rules

- SIEM correlation rules share a lot of the same challenges as analytics
 - In fact, they **are** analytics! Imagine!
- Future vision / desire is for SIEM vendors to support SCO Pattern as a method to define rules
 - Reduce / eliminate vendor lock-in
 - Enable broader ecosystem of cross-vendor solutions sharing tools
 - Seamless integration of STIX 2.0 compatible threat intelligence with SIEM correlation engines
- Again, speak to your vendor!
 - Nothing moves ahead without customers demanding it



It's not perfect...yet.

- Known gaps in SCO object model itself
- Known gaps in language
- We need your help!
- While we believe that STIX Patterning is amongst the most long-term significant innovations in STIX 2.x, it is nevertheless a work product coming out of a very small team of people. If we have succeeded in convincing you that we are not in fact smoking crazy goat-weed, please come join the party!



tl;dr

Thank you!

- Make sure to grab a quick reference card.
- We're having a ½ day STIX/TAXII 2.0 training followed by a ½ day hackathon Friday where you can learn more and try out the tools we discussed.
- Kudos to our colleagues from CIRCL for being so supportive and for early adoption in MISP.
- Thanks to FIRST and OASIS for making this event happen and to you for giving us your attention today!