Meeting Date: Thursday, May 12 2016
Time: UTC 14:00:00
Purpose: CTI-TC CybOX Subcommittee Working Session

Attendees:

Ivan Kirillov - Moderator
Trey Darley
John Wunder
Rich Piazza

John Anderson
Dean Thompson
Bret Jordan
Kyle Maxwell

Greg Back
John-Mark Gurney
Jane Ginn - Recorder

Other Guests

Agenda:

- Object relationships
  - Separate structure vs. embedded referencing
    - See: https://docs.google.com/document/d/1PSGv6Uvo3YyrK354cH0cvdn7gGedbhJkgNVzwW9E6A/edit#heading=h.aiqqrlmfnau
- Patterning
  - How should patterning be defined in CybOX? Separate spec/work product? Part of the 3.0 release?
    - See: https://docs.google.com/document/d/1PSGv6Uvo3YyrK354cH0cvdn7gGedbhJkgNVzwW9E6A/edit#heading=h.f3n8zy5cd1zd
- String field encoding capture
  - See: https://docs.google.com/document/d/1PSGv6Uvo3YyrK354cH0cvdn7gGedbhJkgNVzwW9E6A/edit#heading=h.wpkylnyrvhz1

Meeting Notes:

Object relationships

Ivan

Let’s tackle the issue of how Object Relationships could be done

Main type types to keep:
  - Container Relationships
  - Contextual Relationships – Better than having everything embedded

Biggest question is how to create the Relationships

Now to be a separate structure in the container/ArgleBargle

Have to parse separately

Up-side is that it gives us the ability to handle

-- So, we looked at what the STIX SC

Discussed the option of embedding everything – gave visual example of codes
Discussed other option – Separate relationships

This option does make the objects more complex

We had some discussion on Slack – others seemed to like this approach

Embedding facts – that seems like the way to go... but resolved within the group

Asked for feedback
**John-Mark**

I prefer the embedded type approach

_Gave rationale – Liked separate object_

If you do want to extend the relationships – Can add a custom field

**Ivan**

I think you are right – add an X

First point brought up a good point – Now, relationships only defined by objects –
Don’t go to fields

_Gave example_

In Windows... want to refer to the DLL

_Showed example_

Maybe what we need to do is more write-up on this embedded approach

**Greg**

Brought up a concern about load balancing

**Bret**

Facts should be done as embedded references
Extensions should be done as relationships

**Ivan**

Asked if people thought it should be a list

**John-Mark**

Asked that it be all a Ref or all a string

**Ivan**

Then we would have to make them all Arrays

**John-Mark**

If we have certain Relationships – Would still have to do a check. Is this a list or array?

**Greg**

One more question – Certain Relationships where time matters. Example Domain to IP
Can change over time – Is that a higher level item

**Bret**

That would be at a higher level

**Trey**

Would be done at the Observation level with a Timestamp

**Ivan**

I’m not hearing any objections – We’ll rewrite the text
We’ll have to work through the list of the kinds of relationships
 Asked everyone to look through the list

**Patterning**

**Ivan**

Where should patterning live? STIX or CybOX?
Tied to CybOX Object Model
Outlined two possibilities –

**John-Mark**

I originally thought it would be CybOX because of the Time component...
Now I see that it should be a separate component within CybOX
Patterning approach did not have to be in either – in JSON

**Trey**

_Asked for clarification – Ohh, you could use for MAEC_

**Ivan**

Showed example on screen

**John-Mark**

Not CybOX specific – Can be used in other cases
Ivan

I agree with you... Does not have to be tied to a specific version
What if we add a new field – Could have a pattern that was intrinsically tied to a version

John-Mark

That pattern would .. but, not the language itself
This actually shows...that it is a CybOX issue... Patterning lives in the Indicator Object
Asked a question about Versioning between STIX & CybOX

Trey

I submitted a proposal – Do need tight coupling
Gave an approach for iteration with each – to maintain coupling and interoperability

John-Mark

OK, that makes sense – Like a patch level

Trey

Then, the question is, how do we manage in OASIS land?
Process could take a lot of time for Major releases
Could we make the Minor releases more quickly through the process?

Ivan

That was my main concern with that process

Bret

It will be OK to push to Committee Draft – That is within the TC
All IP issues are locked in place
Then, we could push out to full OASIS – then for Major releases

John-Mark

Yea... OK
So, it does not matter to me if it is inside the Spec

Ivan

Talked about how Patterning would need to Rev, relative to a Major version

John-Mark

I would argue that if you majorly change patterning then you would doing a new CybOX Rev

Ivan

How about for Rev. 3.0, why don’t we just make Patterning part of CybOX... see how it goes
If we need to break it out separately, we can in a subsequent release

NEXT TOPIC

Brought up issue of How to Capture Pattern

Trey

Why don’t you say in Draft Spec – An ArgleBargle can contain either a Relationship or...

Bret

I think it should just be a String

Trey

Came about because Relationship between Indicator and a Sighting

Bret

ArgleBargle concept is between the Sighting and the Container

Trey

That seems to muddy the tear line – *Explained where the ArgleBargle can be used*
If we iterate it would cross over into STIX – Or it would cross into CybOX

John-Mark

I would disagree – STIX pattern is described by the CybOX pattern

John

I was saying in the side-bar – Have an Indicator field... *explains*

Ivan

The only reason I created the CybOX Pattern type
It would be a Pattern String – Showed how it would conform to the spec

John-Mark
Flat is better than nested

Bret
As long as there is not additional Metadata – In the future, we can evaluate that
If we use it this way, any higher level language can use

Ivan
We can define the syntax for the string –

Bret
They have already done that with all of the examples that they developed
So if needed in STIX, would be dumped in

Ivan
OK, I guess we will go with the String definition

John-Mark
Please make a note in the document about what we just talked about – don’t want to lose

NEW TOPIC

Ivan
Where should we put this in the Spec?

Bret
May want to have a Multi-part spec

John-Mark
Jason and I had some more things to do. Let’s keep separate until done

Ivan
I’ll make a note in the document
Anything else on Patterning before we move on

String field encoding capture

Ivan
Anything that can have an encoding that is different... is something you want to specify
To do in CybOX 3.0 we would have to define
So we thought about it... what if we have a sibling field that makes sense...
To capture the encoding of that field
We found this reference from the IANA Registry – but is missing
Preferred MIME type... some are missing – We didn’t have a real agreement
Propose: Reference MIME name (if there) in IANA Registry
If not use the other name

John-Mark
I like, but no way to represent binary data in JSON
This solves another issue – How do we represent binary
If we add a field

John Anderson
You guys are defining – Just state that the user should do

John-Mark
Just a table from IANA – What if IANA versions the table – Break some software

John
I have an answer for this – Just pick the most common

Ivan
We could reference the Version

John
Don’t conflate type with encoding
Ivan
   Right – We are not. This is just type
Trey
   I suspect that the table has not been revisioned
John-Mark
   I disagree – It was revisioned as recently as 2 years ago
               We need to keep a copy – If IANA does rev it
Trey
   What about the RFC? Will IETF still be around?
John-Mark
   I do like approach
Ivan
   Asked how to capture
John-Mark
     Noted that this is more of a JSON serialization issue than it is a Spec issue
               We don’t want to get into this type of detail in the Spec
Trey
   No one will argue with you
Ivan
   Base 64 encoded
John
   We are conflating
     We are showing UTF8 in the example
               When it comes in, we need to re-encode it
John-Mark
   That is one way. I like to keep the original
John
   Binary data could also be defined by a flag
               When a developer is writing a code – He uses File Name – Base64
John-Mark
   I’m a Free BSD developer – Not standardized
John
   File_8 would be the base name – if have something else... needs to be flagged
               Described a process
               I’m making assumption that we prefer UTF8 if at all possible
Ivan
   There probably are Use Cases for binary data as well
Trey
   Brought us a different Use Case – Different character sets
               That is the main reason I’ve heard for dealing with this
Ivan
   Good point – Primary Use Case around this
               Went over John’s proposal
John
   I think the bin flag could be misinterpret
               Gave example of how it could be confused
               Personally I would prefer the full word Encoding
               I’m not going to be typing... Longer tags might be better in the long run
Trey
   This is an interchange format... this is not for human readable
               Might have performance issues
John-Mark
  Most traveling over TAXII – TLS turned off
Ivan
  What does everyone else think?
John
  Are they one or the other – Mutually Exclusive?
  [Some discussion of how to clarify John’s approach – Speaking about encoding format]
Trey
  Have to cut the meeting... and go to another
Ivan
  Thanks everyone

Meeting Terminated