STIX Block #4: Focused Issues

January 15, 2016

Session objective

- Focused discussion on specific sticking points of issues with near-consensus
- For each proposal/issue:
  - VERY briefly describe apparent consensus
  - Confirm/Refute that consensus holds among F2F attendees
  - VERY briefly summarize area of non-consensus
  - Proponents of each option briefly explain their option
  - Open discussion focused on the proposal options for areas of non-consensus
  - VERY briefly summarize outgoing consensus or status
  - Identify outgoing action items

Agenda

- Simplified CV structure (30 mins)
- Sightings (40 mins)
- ID format (35 mins)

Simplified CV structure (30 mins)

- Quick summary of consensus
  - Ability to assert a default controlled vocabulary for a given string property
  - Ability to assert a value for a string property from a specific default set of values (default controlled vocabulary)
  - Ability to assert a value for a string property from a specific non-default set of values (other controlled vocabulary)
    - Should be able to assert an identifier for the other controlled vocabulary
  - Ability to assert both a value for a string property from a specific default set of values and a value for a string property from a specific non-default set of values
  - Ability to assert an arbitrary value for a string property

- Any objections to this consensus?

Simplified CV structure

- Consensus solution
  - We have two types for a controlled vocabulary field
    - Closed - We have an default vocabulary defined in the spec and you can use it or other
      - If you use other then you can use the extension for a defined controlled vocabulary
    - Open - We do not have an default vocabulary defined yet so you can use any string.
      - If you use other then you can use the extension for a defined controlled vocabulary

Simplified CV structure examples

- Closed value, where you use something from the default controlled vocabulary

```
{
  "type" : "indicator",
  "indicator_type" : "malware"
}
```

- Closed value, where you use some arbitrary value with a fallback

```
{
  "type" : "indicator",
  "indicator_type" : "malware",
  "indicator_type_ext" : {
    "value" :              "malware-super-duper"
  }
}
```

Simplified CV structure examples

- Closed value, where you use a defined vocabulary of your own

```
{
  "type" : "indicator",
  "indicator_type" : "other",
  "indicator_type_ext" : {
    "value" : "malware-super-duper",
   "vocab" : "1234-abc-1234-uri"
  }
}
```

- Closed value, where you use a defined vocabulary of your own with a fallback

```
{
  "type" : "indicator",
  "indicator_type" : "malware",
  "indicator_type_ext" : {
    "value" :    "malware-super-duper",
    "vocab" : "1234-abc-1234-uri"
  }
}
```

Simplified CV structure examples

- Closed value, where you use some arbitrary value

```
{
  "type" : "indicator",
  "indicator_type" : "other",
  "indicator_type_ext" : {
    "value" : "malware-super-duper"
  }
}
```

- Open value, you can use any value

```
{
  "type" : "indicator",
  "indicator_type" : "foo bar"
}
```

Simplified CV structure examples

- Open value, you want to use a controlled vocabulary of your own

```
{
  "type" : "indicator",
  "indicator_type" : "other",
  "indicator_type_ext" : {
    "value" : "malware-super-duper",
   "vocab" :  "1234-abc-1234-uri"
  }
}
```

- Open value, where you use a defined vocabulary of your own with a fallback

```
{
  "type" : "indicator",
  "indicator_type" : "malware",
  "indicator_type_ext" : {
    "value" : "malware-super-duper",
   "vocab" :  "1234-abc-1234-uri"
  }
}
```

Simplified CV structure

- Open Discussion
- Outgoing consensus? Status?
- Action Items?

Sightings (RM1) (Prop11) (40 mins)

- Quick summary of consensus
  - Sightings should be broken out of Indicator into a separate “top level” object
  - Sightings should reference an Indicator that was sighted
  - Sightings should enable capturing attribution, time sighting occurred and confidence in the sighting
  - Should have some ability to associate particular observation (what was “seen”) to a Sighting
- Any objections to this consensus?

Sightings (RM1) (Prop11)

- Quick summary of non-consensus areas
  - Should Sighting be a relationship from an observation to an Indicator (even if observation doesn't exist) or an assertion to an Indicator with the option of using a relationship to associate an observation to the sighting if one exists?
  - Should there be a tightly constrained subclass of Sighting to explicitly support very minimal "+1" type sightings?

Sightings (RM1) (Prop11)
Round1 strawman proposals

- I have long asserted that Sightings are a relationship from an Observation to an Indicator.
  - I think I convinced Terry of this leading to the TWIGS approach. ☺

- Considering Sightings in the context of the new relationship objects and the diverse sighting use cases led to an understanding that they are better represented as one-ended Assertions than two-ended assertions (Relationships)

- Solution
  - Sighting object as a type of Assertion with an additional required "Sighted_Indicator" field and optional Count field
  - QuickSighting type derived from Sighting type locked down with Title, Description, Count and Confidence prohibited

Sightings (RM1) (Prop11)
Round1 strawman proposals

- +1 quick sighting


```
{
    "id": "example:sight-b686e902-3ee1-4d7e-a54f-448ccd741651",
    "type": "quick-sighting",
    "timestamp": "2015-12-21T20:49:08.000000+00:00",
    ”sighted_indicator": "example:ind-b8e37090-5d62-45a1-ac2e-a88601b08432"
}
```


- Simple Sighting


```
{
    "id": "example:sight-cbc7d591-87f1-4765-a15e-c39e7b9d12ba",
    "type": "sighting",
    "timestamp": "2015-12-21T20:49:08.000000+00:00",
    " sighted_indicator ": "example:ind-b8e37090-5d62-45a1-ac2e-a88601b08432",
    "count": "2",
    "confidence": “High”
}
```

Sightings (RM1) (Prop11)
Round1 strawman proposals

- Simple Sighting with Observation


```
{
    "id": "example:sight-cbc7d591-87f1-4765-a15e-c39e7b9d12ba",
    "type": "sighting",
    "timestamp": "2015-12-21T20:49:08.000000+00:00",
    "sighted_indicator ": "example:ind-b8e37090-5d62-45a1-ac2e-a88601b08432",
    "count": "2",
    "confidence": “High”
}

{
    "id": "example:obj-fa7de07b-ee02-4773-8cac-fa7161a7f998",
    "type": "file-object-observation",
    "timestamp": "2015-12-21T22:45:12.000000+00:00" ,
    "title": "Suspicious File",
    "hashes": [
        {
            "type": "md5",
            "hash_value": "3773a88f65a5e780c8dff9cdc3a056f3",
        }
    ]
}

{
    "id": "example:rel-9d0c539e-a874-42c7-a055-3e900b98724f",
    "type": "related-observation",
    "timestamp": "2015-12-21T19:59:12.000000+00:00",
    "confidence": “High”,
    "from": "example:sight-cbc7d591-87f1-4765-a15e-c39e7b9d12ba",
    "to": "example:obj-fa7de07b-ee02-4773-8cac-fa7161a7f998",
    "relationship_nature": {
        "value": "Sighted As"
    }
}
```

Sightings (RM1) (Prop11)
Round1 strawman proposals

- In vast majority (likely > 95%) of uses Sighting will have no Observation defined and therefore leave a phantom “From” on a relationship
- A single-ended Assertion ”on/against” an Indicator makes more sense, is simpler and less ambiguous.
- Producers should be able to assert counts of sightings. Count is often as meaningful as simply knowing that party sighted the Indicator.
  - There is still an open question as to how to explicitly clarify what kind of count is provided.
- Allows relating multiple Observations to a Sighting while tracking Confidence in the Sighting and each Observation relationship separately

Sightings (RM1) (Prop11)
Round1 strawman proposals

- Benefits
  - Can assert and/or send just the Sighting alone
  - Can assert of count of how many times it was “seen”
  - Supports both general Sightings as well as tightly constrained +1 Sightings
  - No optional “from” field required
  - No phantom Observations needed
  - Clean, concise, consistent

STIX ROADMAP #1:

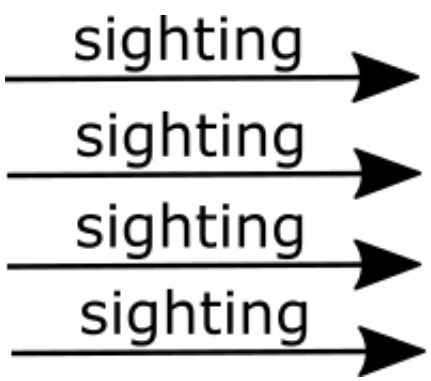
SIGHTINGS

SIGHTINGS

Sightings are just relationships between an Indicator and an Observation

BENEFITS OF SIGHTINGS RELATIONSHIPS

Can send only the relationship to **reduce bandwidth** or **obfuscate** the details.



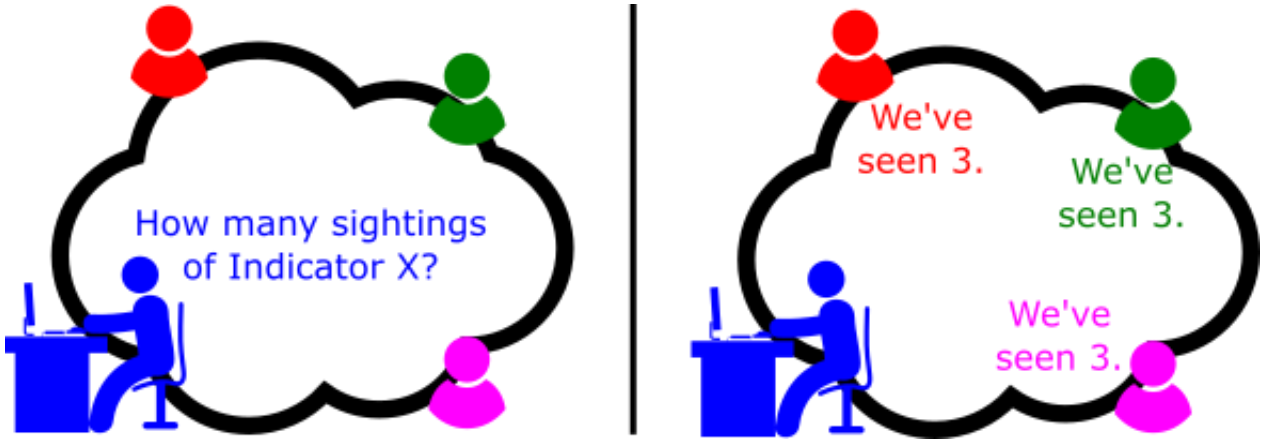
Consumers can subsequently request details of the referenced objects if they wish.

example


```
{
  "type": "announcement",
  "id": "example.com:announcement:69617db3-168a-4010-84a5-bbe622ccfe59",
  "created_at": "2016-01-11T14:11:53.458183-05:00",
  "producer_ref": "example.com:information-source:example",
  "relationships": [
    {
      "type": "relationship",
      "id": "example.com:relationship:61b47ab3-3ac2-4c5a-9ee8-d6680b454c7c",
      "created_at": "2016-01-11T14:12:28.288225-05:00",
      "producer_ref": "example.com:information-source:example",
      "value": "sighted",
      "source_ref": "example.com:observation:8e3de12b-fd0b-4171-8cd5-07d7942eb278",
      "target_ref": "example.com:indicator:dd19ad3e-b027-44de-a196-d0a8b31bfbbb"
    }
  ]
}
```

But what about counts?

Consumers can send **TWIGS requests** to ask for sighting counts, and community members can send **TWIGS responses** containing the counts.



Consumers can subsequently request details of the referenced objects if they wish.

Keep counts out of the data model

Counts are not data: they're a calculation performed on data

Counts complicated the semantics and require special handling

Sightings (RM1) (Prop11)

- Open Discussion
- Outgoing consensus? Status?
- Action Items?

ID format (35 mins)

- Quick summary of consensus
  - Current Qualified Name ID format is undesirable
  - ID should convey identity of producer
  - ID should convey some unique identifier

- Any objections to this consensus?

ID format

- Quick summary of non-consensus areas
  - What is producer-identifiable portion?
    - Domain name?
  - Should identifier include clue to object type?
  - What is unique identifier portion?
    - GUID/UUID?
    - Hash?
    - Should it be flexible to use or formally bound?
  - Should identifier be URI format?

STIX ROADMAP #6:

ID Format

MAKE IDs non-QName

- No messy namespace prefix/full namespace resolution
- Use specially formatted strings vs. URI:
  - Tie IDs to the producer domain name to enable consumers to ask for object details
  - Allow producers to move their TAXII Servers where they want (no FQDN or protocol in ID)
  - Include object type in IDs to better support relationships on their own

What IDs look like

Its actually free-text!

[producer domain name]:[object_type]:[unique id]

e.g.

example.org:indicator:c8b0dfbd-b502-4376-9cec-890ec1a4272b

OR

lockheedmartin.com:kill-chain:lmco-killchain

OR

fox-it.com:threat-actor:business-club

What WE Envisage

We see a future where there are **libraries** of **reusable TWIGS objects** across the world!

We want to encourage producers to leverage each other's Objects.

lockheedmartin.com:kill-chain:lmco-killchain

Human-readable Object IDs will help that happen

Why not a uri?

Ability to embed more information in IDs

Simpler format

Does not tie ID to a protocol

Human-readable Object IDs will help that happen

ID format

- Open Discussion
- Outgoing consensus? Status?
- Action Items?