STIX Block #3: Focused Issues

January 14, 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

- Define consistent Timestamp format (25 mins)
- Relationships (40 mins)
- Source reference for each construct embedded or via relationship (30 mins)

Define consistent Timestamp format

- Quick summary of consensus
  - We need a consistent format for all timestamps
  - We need a precision field
  - Informal consensus recently seemed to be to use RFC 3339 with microsecond precision and timestamp offset (+/- from UTC, not Z)
- Any objections to this consensus?

Define consistent Timestamp format

- Quick summary of non-consensus areas
  - Should the default precision be nanoseconds?
  - Should we use UNIX epoch or RFC 3339 string?
  - If we use RFC 3339 string should we further constrain the string format?
- Would anyone like to champion a position?
- Open Discussion
- Outgoing consensus? Status?
- Action Items?

Relationships (RM4) (Prop9)

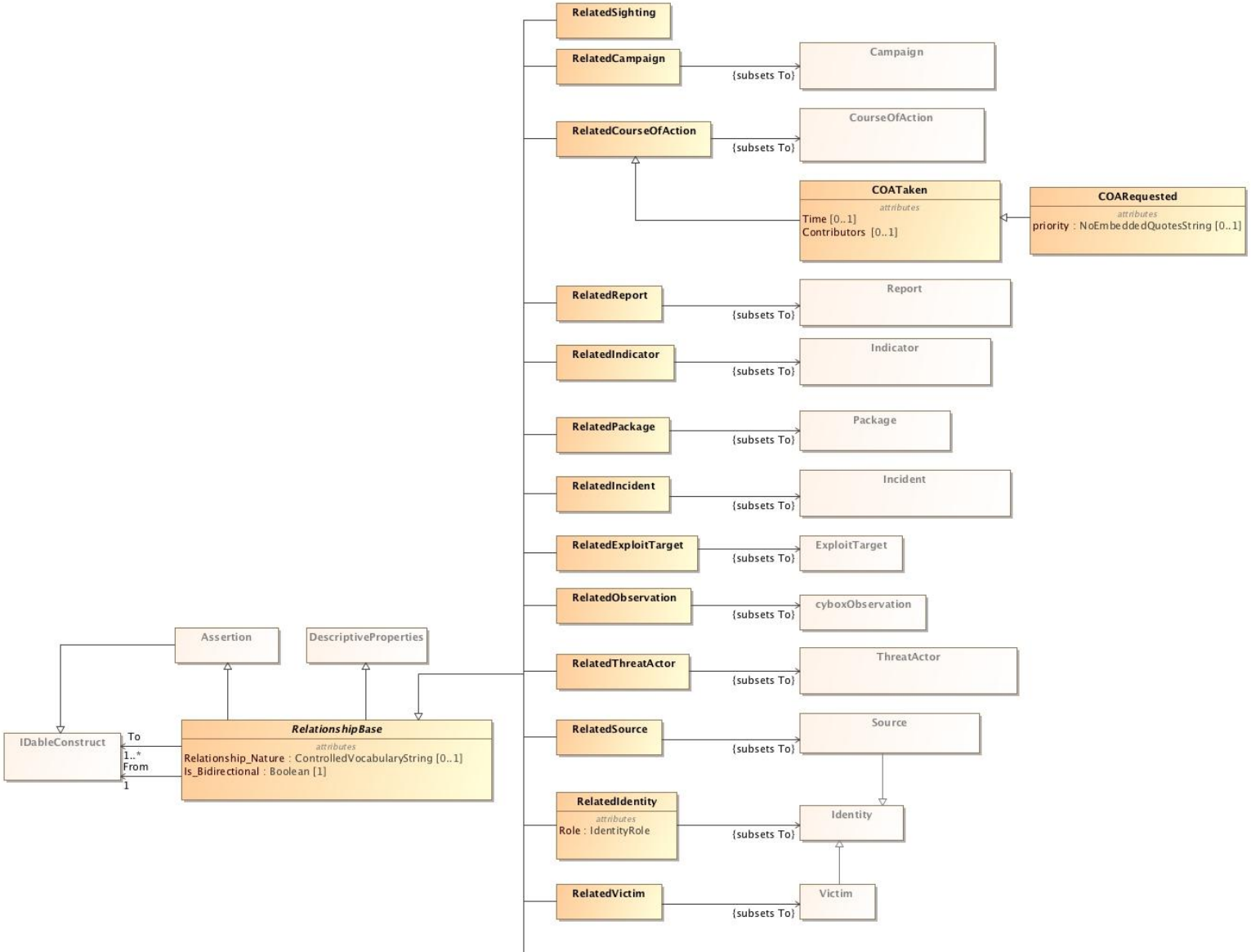
- Quick summary of consensus
  - Relationships should be broken out from other objects and be separate “top-level” objects
  - Relationships should specify both a ”from” and ”to” IDable construct
  - Relationships should support a controlled vocabulary characterization of the nature of the relationship
  - Relationships should support an asserted confidence

- Any objections to this consensus?

Relationships (RM4) (Prop9)

- Quick summary of non-consensus areas
  - Should Relationships be allowed from any IDable construct to any IDable construct?
  - Should subclasses be defined for particular kinds of relationships?
  - Should relationships be only unidirectional or support optional specifying as bidirectional?

Relationships (RM4) (Prop9)
Round1 Strawman Proposals



Relationships (RM4) (Prop9)
Round1 Strawman Proposals

- Generality of relationships
  - Allowing any-to-any relationships sounds easier for analysts but in reality it loses much of the domain semantics and can degrade analytic value of information by short-circuiting important interim relationships
    - For example, it could be very tempting to assert a relationship from an Indicator to a ThreatActor (TA). In reality such a belief/assertion is almost always based on the Indicators relationship to particular TTP that has also been asserted as related to the TA. By allowing the “easy” short cut the truly useful information has been lost.
      - You don’t necessarily know why the Indicator is asserted as related to the TA. “Short cut” relationships lead to low confidence relationships.
      - If a new TA is discovered using the TTP you don’t have the connections to make the adjustment.
      - If new info arises that the TTP is no longer used by the TA you don’t have the connections to make the adjustment.

Relationships (RM4) (Prop9)
Round1 Strawman Proposals

- Generality of relationships
  - Information analysts interested in “easy” way to do first order analysis will often love this.
  - Intelligence analyst who rely on quality and richness of information will often hate this.
  - Requires implementation to always parse into relationship and interpret string field to determine if relationship type is supported and what it means.
  - May require fairly large single vocab for relationship nature

Relationships (RM4) (Prop9)
Round1 Strawman Proposals

- Relationship subclassing
  - Much simpler for implementations to parse (does not require parsing into object and interpreting string field to determine if relationship type is supported)
  - Enable support for properties specific to particular types of relationships
  - Explicitly binds the semantics of which IDable construct types can be involved in the relationship
  - Explicitly clear to any consumer what type of relationship it is
  - Allows for focused relationship nature vocabs rather than single large vocab across all relationships

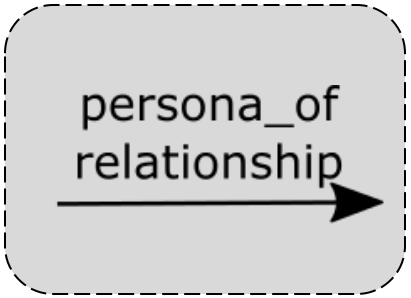
Relationships (RM4) (Prop9)
Round1 Strawman Proposals

- Bidirectionality of Relationships
  - Some CTI relationships are unidirectional (e.g. versioning derivation) while others are bidirectional (e.g. Incident attribution)
    - Need to support both types
  - Implicitly presuming directionality from nature is ambiguous and risky to think about in a graph nature
    - Explicit is better than implicit
  - Requiring bidirectional relationships to be defined separately in each direction is more complex, more verbose and more prone to maintainability and consistency issues
  - Simple to just have an Is_Bidirectional boolean property on relationships to assert if relationship is unidirectional or bidirectional

STIX ROADMAP #3:

RELATIONSHIPS

RELATIONSHIPS are METADATA OBJECTS



THIRD PARTY RELATIONSHIPS

Two pre-existing Domain Objects

Third-party Relationship relates them together

Twigs defines relationships

- TWIGS defines well-known, meaningful relationships for common CTI use cases

But doesn't limit you

- …but allows tools/analysts to create additional relationships that we haven't thought of

Descriptors should be meaningful

- “related-ttp” doesn’t give you additional information
- Focus on data points that add information

directionality

- Don’t mix undirected and directed edges (relationships)
  - The relationship value can tell you whether you should consider the other direction
- Bidirectional relationships add complexity and cognitive overhead to distinguish edge cases

Relationships (RM4) (Prop9)

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

Source reference for each construct embedded or via relationship (RM3, 4 & 7)

- Quick summary of consensus
  - Information source is a separate Source object
  - Need ability to associate an IDable construct with a Source object
  - Source object leverages Identity construct

- Any objections to this consensus?

Source reference for each construct embedded or via relationship (RM3, 4 & 7)

- Quick summary of non-consensus areas
  - Approach #1: Source association asserted via Relationship from IDable construct to Source with nature of “Has Source”
  - Approach #2: Source association asserted via Producer_ref property embedded within the IDable construct.

Source reference location (RM3, 4 & 7) Round1 Strawman Proposals

- Consistently leverage the “one way of doing things” for relating objects to each other.
- Enable Confidence and other Relationship properties useful for asserting uncertain or non-originating Source relationships (Source for content is not always producer self-asserted)
- Enables 3rd party assertions of Source attribution
- Creating a many-to-one Has Source relationship could significantly improve verbosity of content (especially large bodies)
- When same exact content received from multiple sources, allows you to characterize (with confidence) them separately

STIX ROADMAP #3, #4, #7:

References
and relationships

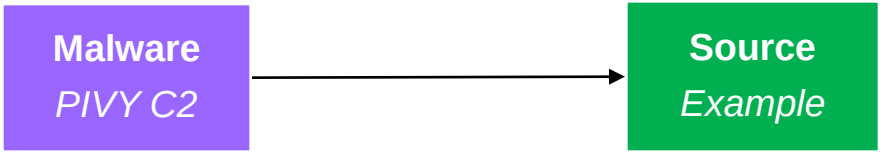
TWIGS USES REFERENCING

Top-level objects are NEVER embedded in other top-level objects

DOMAIN Objects ARE REFERENCED Via relationships

```
{
  "type": "relationship",
  "id": "example.com:6d8a1ef3-3b9b-4241-8f57-0f25b5db52d5",
  "created_at": "2016-01-06T19:16:06.357578-05:00",
  "producer_ref": "example.com:example",
  "source_ref": "example.com:628a75ab-7b63-4e88-9175-2c190e38233c",
  "target_ref": "example.com:52c6211a-ab03-4fab-b613-f78ffbf8059f",
  "value": "indicated",
  "credibility": 4
}
```

METADATA OBJECTS and domain objects reference each other directly by id

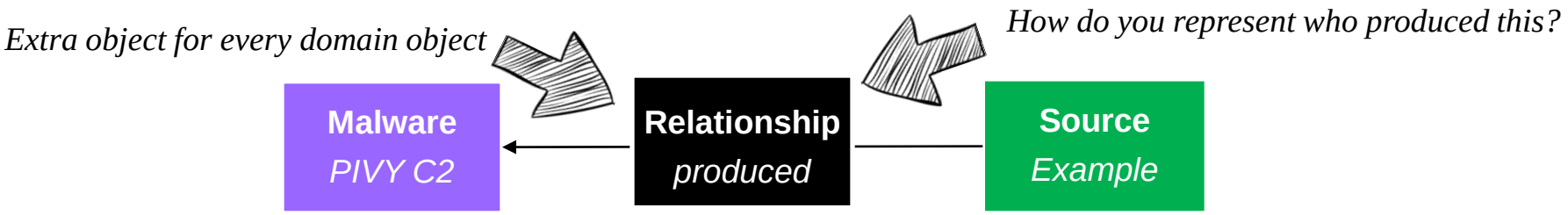



```
{
  "type": "malware",
  "id": "example.com:52c6211a-ab03-4fab-b613-f78ffbf8059f",
  "created_at": "2016-01-06T19:15:32.963619-05:00",
  "producer_ref": "example.com:example",
  "title": "Poison Ivy C2"
}
```

Why?

Simplicity & consistency

- Avoid a proliferation of relationships
  - Each object would require a companion relationship
  - Relationships should be for domain objects
- Easily and consistently represent the source/producer of the relationship itself



Source reference for each construct embedded or via relationship (RM3, 4 & 7)

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