TAXII WORKING CALL

2018-05-29

Agenda

1. Agenda Bashing
2. Status of WD03 / CSD01
3. TAXII Query

Status of CSD01

- Working Draft 03 is ready to be released
- Depending on the timing of STIX 2.1 CSD01, we will either wait or release WD03 as a CSD01 ballot.
  - Want to avoid ballot fatigue

TAXII Query - Features

- Easier / Near-term Needs
- Harder / Long-term Needs
- Scope
- Goals
  - Identify the types of features we need to support
  - Get high level agreement on what to do first
  - Two existing proposals (RESTful design, query object)

NEAR-TERM NEEDS

EASIER FEATURES

Near-term Query Needs

- Find Related Objects By ID
- Dereference Content
- RESTful Requests

Find Related Objects By ID

- Problem
  - Given any object ID, there is no way for a client to ask a TAXII Server for any relationships that connect to that object.
- Proposal
  - TAXII should support the ability to find objects that relate to a given ID, either embedded or external relationship.

Find Related Objects By ID

- For example:
  - Given an Indicator STIX ID
  - Return all relationships that list that ID in one of the fields.
  - Filter by relationship type and which way it is points
  - Give me all objects where this identity is the creator (filtering by an attribute??)
- Could be done via a URL parameter or URL endpoint
  - ?match[type]=relationship&match[relationship]=<id>
  - ../<collection-id>/relationships/related-to/<stix-id>/

Dereference Content

- Problem
  - When a client requests a report or another object with embedded references, it will have to make potentially many requests to the server to get all of the objects identified in the report.
- Proposal
  - TAXII should support the ability to automatically dereference embedded relationships.

Dereference Content

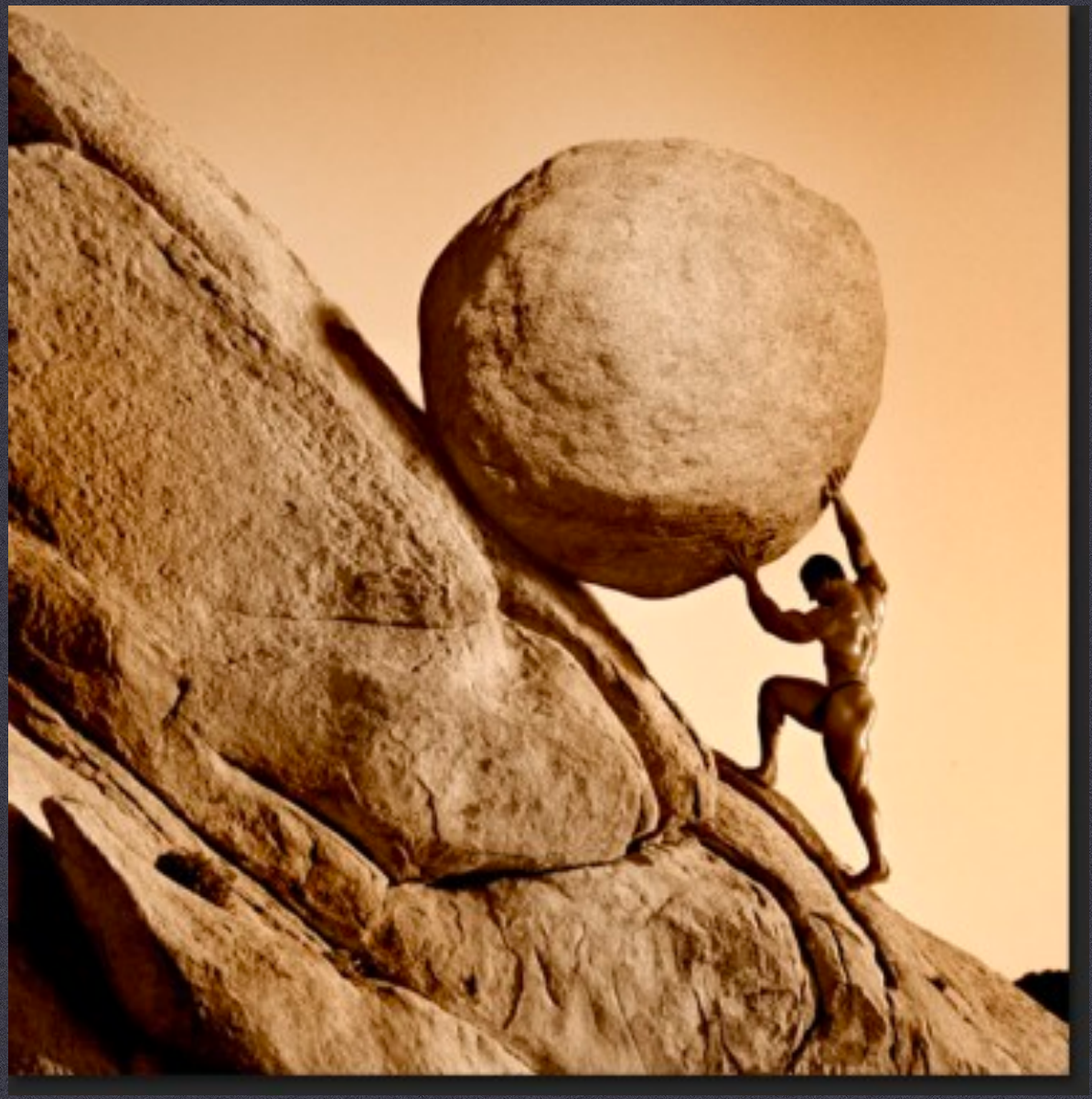
- For example:
  - Given a request for a STIX indicator
  - The client should be able to say auto-dereference the object
  - Then the TAXII server should send the identity object that is linked via the created_by_ref at the same time.
- Could be done via a URL Parameter, for example:
  ?ref=1, ref=yes
- Should an error be returned if something can not be found or not returned
- How deep should it go?
- Risks of accidentally requesting to much data, like if you ask for an identity and you get all objects created by that identity. Need to figure out a solution for this.
  - May solve this by only allowing one way

RESTful Requests

- Problem
  - Users of the CTI ecosystem are wanting an easier way of requesting specific objects from a TAXII Server in a more RESTful way.
- Proposal
  - TAXII should support more REST like feature in the URL path.

LONG-TERM NEEDS

HARDER FEATURES



Long-term Query Needs

- Query Based on a Pattern
- Query Based on an Observed Data
- Query Based on Properties
- Graph Traversal Queries
- Resolve To Some Depth

Query Based on Pattern

- Problem
  - Threat researchers need the ability to query a TAXII Server to find observations or other data based on some STIX Indicator Pattern
- Proposal
  - The pattern probably needs to be sent in a query resource. This would mean that we would natively be using STIX concepts in a TAXII resource.
- We may need to do levels of support for this

Query Based on Observed Data

- Problem
  - Users need the ability to query a TAXII Server to discover if any STIX Indicators exist based on some Observed Data (Observation)
  - Need ability to discovery any observations or other data based on some Cyber Observable data.
- Proposal
  - The observation probably needs to be sent in a query resource. This would mean that we would natively be using STIX concepts in a TAXII resource.
- This might be pushing SIEM features in to TAXII, we may need to do levels of support for this as well.

Query Based on Observed Data

- Examples
  - query: /api/v0.1/observable/win_registry/value/<value>/ (where value is a valid windows registry value)
  - query: /api/v0.1/observable/domain_name/<value>/ (where value is a valid FQDN or TLD)
  - query: /api/v0.1/observable/address/ip/<value>/ (where value can be a single IP, an IP range, or a CIDR block)

Query Based on Properties

- Problem
  - Threat researchers need the ability to query a TAXII Server for objects that contain or are like a certain amount of data.
  - Given some data from some STIX object properties, find objects with a certain combination of properties
- Example
  - Given some virus total or sandbox results, return all Malware objects that have these properties.
- How to prioritize the results you get back, sighting time stamps, indicator validity, etc

Graph Traversal Queries

- Graph traversal queries (ie tell me if A and B are connected, and if so, return the path(s))
- How we capture what they want to traverse, I can draw a line between these objects but I can not return the object. TLP restrictions or I was never actually given the object so I can not give it to you.

Resolve to Some Depth

- TAXII should support the ability for a client to tell it to automatically send external relationship objects and their end points to some depth level.
- This depth level should be configurable on the server and probably advertised either at the api-root level, server level, or maybe even collection level.

SCOPE

WHERE TO QUERY

Query Scope

- Query Per Collection ID
- Query Across All Collections
  - All that the user has permission to query

Chat Notes:

From John Wunder to Everyone: (01:29 PM)
Another example would be in a world with sightings, you might end up getting a zillion sightings for an indicator when you just wanted the related attack pattern or intrusion set or whatever
I don't know how to comment on this one (RESTful requests)…it's pretty general so hard to comment on
From Jason Keirstead to Everyone: (01:31 PM)
I highly enoucrage everyone to please read and comment on https://docs.google.com/document/d/1Cy_9Bh5tKEkDHGg2iv5c3AwriqVr7ygbKXWOv4-uHxs/edit#
From Jeff Mates (DC3) to Everyone: (01:33 PM)
If we follow one wayedness I don't think that would spiral too terribly. Since you would likely get something like sighting -> malware -> observed data; sighting -> observed data; sighting -> identity.
From John Wunder to Everyone: (01:34 PM)
I was thinking starting with indicator. You go indicator => 1 zillion sightings, when you just wanted indicator => Attack Pattern, Malware, Intrusion Set, etc
of course it's surmountable, but we need to think about the types of queries people will make and kind of TTX what the variety of responses would be
From Jeff Mates (DC3) to Everyone: (01:34 PM)
if we follow one wayedness then you can't (by default) embed sighting into indicator you link them the other way around.
Well what if you want to query that way? IMO we can't just limit the types of queries people will make
So you'd need to follow up with a _ref query looking for every _ref to the indicator
From Jeff Mates (DC3) to Everyone: (01:35 PM)
rather than automatically expanding from the indicator itself
From John Wunder to Everyone: (01:36 PM)
Yeah, we don't really need to solve it here, I'd be happy if it's not an issue.
From Jeff Mates (DC3) to Everyone: (01:36 PM)
The tell me all things that have a relationship to X
From Jason Keirstead to Everyone: (01:36 PM)
@Jeff our proposal solves what you're talking about. You can choose to resolve related objects or not, with any query.
From Jeff Mates (DC3) to Everyone: (01:58 PM)
I like the idea of being able to query all collections at once