TAXII

Face 2 Face - Jan 2018

Manifest Resources

Add clarifying text about the date of objects in a collection that have versions #24

Manifest resource and media types #30

Changes to manifest resource #36

Add manifest entry point on objects #37

Manifest Resources

- Current Entry Point
  - /<api-root>/collections/<id>/manifest/
- Current Properties
  - id (required) - string
  - date_added (optional) - string
  - versions (optional) - list of type string
  - media_types (optional) - list of type string

Manifest Resources - Problems

- There are two problems with the manifest resource that we need to talk about

Manifest Resources - Problems

- First: Media Types are not bound to a version as objects get sent around and updated, you can get in to a situation where you have the below and that means you have no way of knowing which version is in which format.
  - indicator--1 ver 0 in STIX 2.0 format and
  - indicator--1 ver 1 in STIX 2.1 format

Manifest Resources - Problems

- Second: Pagination issues when you have versions that get added after a bunch of other records. What do you include in the versions field? What happens when an object has been versioned a LOT.
  - indicator--1 ver 0
  - indicator--1 ver 1
  - indicator--2 ver 0
  - indicator--3 ver 0
  - indicator--1 ver 2

Manifest Resource - Proposed Solution

- Keep current entry point
  - /<api-root>/collections/<id>/manifest/
- Change properties to represent atomic objects
  - id (required) - string
  - date_added (optional) - string
  - versions (optional) - string
  - media_types (optional) - string

Manifest Resource - Proposed Solution

- Add another entry point
  - /<api-root>/collections/<id>/objects/<id>/manifest/
- This resource will return a single object but with all of the versions that the server knows about

Pagination

Get Object by ID needs to support pagination #21
We should probably drop the concept of items for pagination #23

Pagination

- We currently have the concepts of "items" and are using the HTTP Range headers
- The server can force pagination, even when the client does not request it, which is a violation of HTTP (we call that out in the spec)
- We tried to be as vanilla HTTP as possible despite issue above
- There were no use cases driving a specific pagination design, we just knew we needed some form of pagination and just picked one

Pagination - Problem

- The concepts of items only valid within a given instance of a single request. There is no guarantee that the server will have the same data associated with the same item numbers for a consecutive query.
- After implementing this several different ways, I do not see any real value in an items based pagination strategy.
- In fact, all it does is impose a lot of load and performance issues along with implementations complexity, this is also apparently a known bad design for RESTful APIs

Pagination - Proposal

- Change pagination to only use the "date added" concept. This will make it super easy and performant on large datasets
- Specify that if no date filters are provided, then the most current records are returned
- Specify that pagination can happen on the object-id URL

Pagination - Proposal

- Add a URL parameter of "limit" for the client to tell the server how many records it wants.
- Add X-Headers to give optional counts of collection size and number of records requested / returned:
  X-TAXII-Collection-Count (Optional)
  X-TAXII-Response-Limit (Required)

Discovery

Discovery

- We have a hard coded entry of /taxii/
- We needed to define the URL since DNS SRV records can only contain an IP address
- This was done to help a client be able to auto find the TAXII server

Discovery - Ideas

- Redefine the fixed URL
- NAPTR records in DNS, this might work
- DNS-SD, this might work
- TXT records
  - To redirect foo.bar.com to foo2.bar.com/path, just add foo IN TXT "1|foo2.bar.com/path" in your bar.com DNS zone.
  - It also keeps the url paths and parameters. So if you try to access foo.bar.com/hello?foo=bar, you'll be redirected to foo2.bar.com/path/hello?foo=bar.

API-Root URLs

API-Root URLs

- The current specification is not really clear if relative path URL are allowed.
- All examples use fully qualified URLs
- The idea during the writing was that they should be fully qualified URLs
- However, this is problematic for hosted services
- Need to avoid path traversal vulnerabilities

Media Types

Need to change the media type for TAXII per OASIS / IETF / IANA #29

Media Types

- Right now we have two defined for STIX and TAXII
- application/vnd.oasis.taxii+json
- application/vnd.oasis.stix+json
- We need to use the standard tree not the vendor tree before IANA will approve these.
- Confusion around certain endpoints being STIX media types and others being TAXII

Media Type - Proposal

- Combine the STIX and TAXII media types into one media type for IANA.
  - application/taxii+json
- This will simplify interactions for common data, but will not prevent users from delivering other content at object endpoints if they want.

Other HTTP Methods

Deleting objects from taxii server collections #38

Add Support for DELETE

- We need the ability for a client to send a DELETE HTTP Method so that a client can delete content that it has sent.
- This is probably an optional feature or a feature that is limited by various authentication
- We should consider changing the can_read and can_write to be "permissions" and then have a dictionary of permissions inside it.

TAXII Query

TAXII Channels