Proposal: Tiered Filtering

Marlon – DHS / Emmanuelle – MITRE
July 16th, 2019
Goals

• Expand filtering capabilities via matching against any object property.

• It would address:
  • The relationship pivoting use case
    • Examples: 1, 2*
  • The ability to filter upon specific information if an ID is not known in advance
    • Examples: 3*, 4, 6
  • Filtering on TLP Markings*
    • Example: 5
  • Filtering on confidence values*
    • Example: 6
  • Identify sighted indicator
    • Example: 7
  • The ability to query internal references
    • Example: 8

*This is a joint AIS and Federal community use-case
What are we proposing...

• A tiered approach such that TAXII implementers can choose what level of filtering they support

<table>
<thead>
<tr>
<th>Level</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T0 (basic)</td>
<td>The current level of Filtering in the spec. Requests that include id, version, type and/or spec_version</td>
</tr>
<tr>
<td>T1 (common-only)</td>
<td>A TAXII Server advertising T1 support must support requests from T0 and any common properties defined in the STIX spec.</td>
</tr>
<tr>
<td>T2 (full-spec)</td>
<td>Supports the above plus, any property present for formally defined STIX2 objects (relationship_type, target_ref, ...)</td>
</tr>
<tr>
<td>T3 (extended)</td>
<td>Supports the above plus, any request that contains custom properties (?match[x_foo_bar]=something)</td>
</tr>
</tbody>
</table>
What do we need...

• TAXII Servers must provide a way to advertise the support implemented by the server
  • We could use the discovery resource for this purpose

```json
{
    "title": "Some TAXII Server",
    "description": "This TAXII Server contains a listing of...",
    "contact": "string containing contact information",
    "default": "https://example.com/api2/",
    "filter_support": "basic",
    "api_roots": [
        "https://example.com/api1/",
        "https://example.com/api2/",
        "https://example.net/trustgroup1/"
    ]
}
```
Error handling

• We propose that error messages should be returned instead of ignoring filter values that a TAXII Server does not understand.
Caveats

• Currently the spec calls for the return of latest versions of each object, but keep in mind the common properties, and STIX2 property name are different between versions (2.0 vs 2.1)
## Types of queries enabled by this proposal

<table>
<thead>
<tr>
<th>Query</th>
<th>Filtering level</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>?match[type]=relationship&amp;match[target_ref]=&lt;identifier&gt;</code></td>
<td>T2</td>
</tr>
<tr>
<td><code>?match[external_id]=CVE-2016-1234*</code></td>
<td>T1</td>
</tr>
<tr>
<td><code>?match[type]=observed-data&amp;match[value]=1.2.3.4</code></td>
<td>T2</td>
</tr>
<tr>
<td><code>?match[object_marking.refs]=marking-definition--34098fce-860f-48ae-8e50-ebd3cc5e41da,marking-definition--f88d31f6-486f-44da-b317-01333bde0b82&amp;match[type]=indicator&amp;match[created_by_ref]=&lt;identifier&gt;*</code></td>
<td>T1</td>
</tr>
<tr>
<td><code>?match[type]=indicator&amp;match[confidence]=90,91,92,93,94,95,96,97,98,99,100*</code></td>
<td>T1</td>
</tr>
</tbody>
</table>
Example 1

- \(?\text{match[type]}=\text{relationship}&\text{match[target_ref]}=\langle\text{identifier}\rangle\)
- \(?\text{match[type]}=\text{relationship}&\text{match[source_ref]}=\langle\text{identifier}\rangle\)
- “Given an ID, I want to know what relationships point to it“ [2][3]
- “Given an ID, I want to know what relationships have it as source“

```json
{
  "type": "relationship",
  "spec_version": "2.1",
  "id": "relationship--df7c87eb-75d2-4948-af81-9d49d246f301",
  "created": "2016-04-06T20:06:37.000Z",
  "modified": "2016-04-06T20:06:37.000Z",
  "relationship_type": "indicates",
  "source_ref": "indicator--a740531e-63ff-4e49-a9e1-a0a3eed0e3e7",
  "target_ref": "malware--9c4638ec-f1de-4ddb-abf4-1b760417654e"
}

{
  "type": "relationship",
  "spec_version": "2.1",
  "id": "relationship--df7c87eb-75d2-4948-af81-9d49d246f301",
  "created": "2016-04-06T20:06:37.000Z",
  "modified": "2016-04-06T20:06:37.000Z",
  "relationship_type": "indicates",
  "source_ref": "indicator--a740531e-63ff-4e49-a9e1-a0a3eed0e3e7",
  "target_ref": "malware--9c4638ec-f1de-4ddb-abf4-1b760417654e"
}
```
Example 2

- \(?\text{match[relationship\_type]}=\text{mitigates} & \text{match[target\_ref]}=<\text{vulnerability--identifier}>\)
- “Get all mitigation relationships that target a specific vulnerability” *

```json
{
    "type": "relationship",
    "spec\_version": "2.1",
    "id": "relationship--df7c87eb-75d2-4948-af81-9d49d246f301",
    "created": "2016-04-06T20:06:37.000Z",
    "modified": "2016-04-06T20:06:37.000Z",
    "relationship\_type": "mitigates",
    "source\_ref": "course-of-action--a740531e-63ff-4e49-a9e1-a0a3eed0e3e7",
    "target\_ref": "vulnerability--9c4638ec-f1de-4ddb-abf4-1b760417654e"
}
```
Example 3

• `match[external_id]=CVE-2016-1234`

• “Get intel on any object that references **CVE-2016-1234**” *

```json
{
    "type": "vulnerability",
    "spec_version": "2.1",
    "id": "vulnerability-0c7b5b88-8ff7-4a4d-aa9d-feb398cd0061",
    "created": "2016-05-12T08:17:27.000Z",
    "modified": "2016-05-12T08:17:27.000Z",
    "name": "CVE-2016-1234",
    "external_references": [
        {
            "source_name": "cve",
            "external_id": "CVE-2016-1234"
        }
    ]
}
```
Example 4

- `?match[type]=observed-data&match[value]=1.2.3.4`

- "Does anyone have intel on IP 1.2.3.4?" [1][6]
Example 5


• “Get all TLP:WHITE OR TLP:GREEN indicators created by a specific entity” *

```json
{
  "type": "indicator",
  "spec_version": "2.1",
  "id": "indicator--a740531e-63ff-4e49-a9e1-a0a3eed0e3e7",
  "created_by_ref": "identity--311b2d2d-f010-4473-83ec-1edf84858f4c",
  "created": "2017-01-01T00:00:01.000Z",
  "modified": "2017-01-01T00:00:01.000Z",
  "indicator_types": [
    "malicious-activity"
  ],
  "pattern": ":[file:hashes.MD5 = 'd41d8cd98f00b204e9800998ecf8427e']",
  "valid_from": "1970-01-01T00:00:01Z",
  "object_marking_refs": [
    "marking-definition--613f2e26-407d-48c7-9eca-b8e91df99dc9"
  ]
}
```
Example 6

- `?match[type]=indicator&match[confidence]=90,91,92,93,94,95,96,97,98,99,100`
- “Get all indicators with high confidence” *

```json
{
    "type": "indicator",
    "spec_version": "2.1",
    "id": "indicator--a740531e-63ff-4e49-a9e1-a0a3eed0e3e7",
    "created": "2017-01-01T00:00:01.000Z",
    "modified": "2017-01-01T00:00:01.000Z",
    "indicator_types": [
        "malicious-activity"
    ],
    "pattern": "[file:hashes.MD5 = 'd41d8cd98f00b204e9800998ecf8427e']",
    "valid_from": "1970-01-01T00:00:01Z",
    "confidence": 91
}
```
Example 7

- \( \text{?match[type]=sighting}&\text{match[sighting\_of\_ref]=<indicator--identifier>} \)
- “I have an indicator and I want to pull all sightings.” [4]

```json
{
   "type": "sighting",
   "spec_version": "2.1",
   "id": "sighting--bfbc19db-ec35-4e45-beed-f8bde2a772fb",
   "created": "2016-04-06T20:06:37.000Z",
   "modified": "2016-04-06T20:06:37.000Z",
   "sighting\_of\_ref": "indicator--a740531e-63ff-4e49-a9e1-a0a3eed0e3e7",
   "where\_sighted\_refs": [
      "identity--311b2d2d-f010-4473-83ec-1edf84858f4c"
   ]
}
```
Example 8

• ?match[type]=report & match[object_refs]=<identifier>
• “Get any report object that contains a link to this SDO/SRO” [5]

```json
{
  "type": "report",
  "spec_version": "2.1",
  "id": "report--84e4d88f-44ea-4bcd-bbf3-b2c1c320bcb3",
  "created_by_ref": "identity--311b2d2d-f010-4473-83ec-1edf84858f4c",
  "created": "2015-12-21T19:59:11.000Z",
  "modified": "2015-12-21T19:59:11.000Z",
  "name": "The Black Vine Cyberespionage Group",
  "description": "A simple report with an indicator and campaign",
  "report_types": [
    "campaign"
  ],
  "published": "2016-01-20T17:00:00Z",
  "object_refs": [
    "indicator--a740531e-63ff-4e49-a9e1-a0a3eed0e3e7",
    "campaign--8e2e2d2b-17d4-4cbf-938f-98ee46b3cd3f",
    "relationship--df7c87eb-75d2-4948-af81-9d49d246f301"
  ]
}
```
Reference to GitHub issues/Slack


2. (G #6) Add ability to find all objects related to a particular STIX object ID, to prevent an indeterminate number of queries to find them all. [https://github.com/oasis-tcs/cti-taxii2/issues/6](https://github.com/oasis-tcs/cti-taxii2/issues/6)

3. (G #7) Need ability to request related objects in one request to a distance of 1(?). [https://github.com/oasis-tcs/cti-taxii2/issues/7](https://github.com/oasis-tcs/cti-taxii2/issues/7)

4. (G #15) As a User, I want to traverse the STIX graph over TAXII in an efficient manner, so I don't waste resources. [https://github.com/oasis-tcs/cti-taxii2/issues/15](https://github.com/oasis-tcs/cti-taxii2/issues/15)
Reference to GitHub issues/Slack

5. (G #68) No way to query internal references.  
   https://github.com/oasis-tcs/cti-taxii2/issues/68