A TAXII Search / Pivoting Proposal

I would propose that we insert a new section between 5 and 6 and call it "TAXII Query". Then in the conformance section, we add this as a TAXII Query Server that would include the functionality of a TAXII Collection Server plus this endpoint.

In the future we could add a POST method to the /search/ part of this endpoint to allow more advanced queries to be encapsulated. But this will give us a framework for simple content queries that are RESTful and follow the existing designs we have in TAXII.

Search Relationship Objects

This Endpoint retrieves relationship objects from a Collection that match the provided filter parameters. Pagination is supported by the `limit` URL query parameter (see section 3.4) and the `more` property of the `envelope` (see section 3.6).

If a client fails authentication then this endpoint **MUST** return either an HTTP 401 (Unauthorized) or an HTTP 404 (Not Found).

If the Collection specifies `can_read` as `false` for a particular client, this Endpoint **MUST** return an HTTP HTTP 403 (Forbidden) or HTTP 404 (Not Found) error.

To support searching the Collection for relationships, this Endpoint supports filtering as defined in this section and in section 3.4. Clients can provide one or more filter parameters to get relationships objects with a specific version or matching specific objects. Future versions of TAXII will add more advanced filtering capabilities.

When requesting STIX 2 content, that content will always be delivered in a TAXII `envelope` even if there only one object returned. If no STIX objects are returned, the response **MUST** include an empty envelope.

If a STIX object is not versioned (and therefore does not have a `modified` timestamp) then any `version` parameter **MUST** use the `created` timestamp. If an object does not have a `created` or `modified` timestamp or any other version information that can be used, then the server should use a value for the version that is consistent to the server.

Endpoint Specifically Supported Fields for Match

If a request does not contain one of the following fields for the match parameter, the server **MUST** return an HTTP 400 (Bad Request).

Match Field	Description
`related_source`	The STIX identifier that matches the source_ref property in a STIX Relationship object.

	Examples
	`?match[related_source]=indicator--3600ad1b-fff1-4c98-bcc9-4de3bc2e2ffb`
	`?match[related_source]=indicator--3600ad1b-fff1-4c98-bcc9-4de3bc2e2ffb`
	`,sighting--4600ad1b-fff1-4c58-bcc9-4de3bc5e2ffd`

`related_target`	The STIX identifier that matches the target_ref property in a STIX Relationship object.
	
	Examples
	`?match[related_target]=indicator--3600ad1b-fff1-4c98-bcc9-4de3bc2e2ffb`
	`?match[related_target]=indicator--3600ad1b-fff1-4c98-bcc9-4de3bc2e2ffb`
	`,sighting--4600ad1b-fff1-4c58-bcc9-4de3bc5e2ffd`

`related_to`	The STIX identifier that matches either the source_ref or the target_ref property in a STIX Relationship object.
	
	Examples
	`?match[related_to]=indicator--3600ad1b-fff1-4c98-bcc9-4de3bc2e2ffb`
	`?match[related_to]=indicator--3600ad1b-fff1-4c98-bcc9-4de3bc2e2ffb,sig`
	`hting--4600ad1b-fff1-4c58-bcc9-4de3bc5e2ffd`

`related_type`	A string value that matches the relationship_type property in a STIX Relationship object.
	
	Examples
	`?match[related_type]=derived-from`
	`?match[related_type]=derived-from,related-to`

`related_object_type`	A string value that matches the type of object that one end of the relationship is connected to. For example, this could be used to return all relationships that have a `related_type` of dropped-by and an object type of file.
	
	Examples
	`?match[related_object_type]=malware`
	`?match[related_type]=dropped-by&match[related_object_type]=file`

GET /{api-root}/collections/{id}/search/relationships/

Implementation Notes

Get all relationship objects from a collection that match the provided filter
parameters.

Requests

URL Parameters

```
{api-root} - the base URL of the API Root
{id}       - the identifier of the Collection being requested

URL Filtering Parameters
added_after  - a single timestamp            (e.g., ?added_after=...)
limit        - a single integer              (e.g., ?limit=...)
version      - the version(s) of an object   (e.g., ?match[version]=...)
spec_version - the specification version(s)  (e.g., ?match[spec_version]=...)

Endpoint URL Filtering Parameters
related_source - a STIX identifier           (e.q., ?match[related_source]=...)
related_target - a STIX identifier           (e.q., ?match[related_target]=...)
related_to     - a STIX identifier           (e.q., ?match[related_to]=...)
related_type   - a STIX relationship type    (e.q., ?match[related_type]=...)
related_object - a STIX object type          (e.q., ?match[related_object]=...)

Required Headers
Accept: application/taxii+json;version=2.1
```

Successful Responses

```
Response Codes
200 - The request was successful

Required Headers
Content-Type: application/taxii+json;version=2.1
X-TAXII-Date-Added-First: timestamp
X-TAXII-Date-Added-Last: timestamp

Payload
envelope
```

Failure Responses

```
Response Codes
400 - The server did not understand the request or filter parameters
401 - The client needs to authenticate
403 - The client does not have access to this objects resource
404 - The API Root or Collection ID are not found or the client does not have
access to this resource
406 - The media type provided in the Accept header is invalid

Required Headers
Content-Type: application/taxii+json;version=2.1

Payload
error
```

Example

GET Request
```
GET
/api1/collections/91a7b528-80eb-42ed-a74d-c6fbd5a26116/search/relationships/?ma
tch[related_to]=indicator--3600ad1b-fff1-4c98-bcc9-4de3bc2e2ffb&added_after=201
8-02-07T08:44:12.312451Z HTTP/1.1
Host: example.com
Accept: application/taxii+json;version=2.1
```

GET Response
```
HTTP/1.1 200 OK
Content-Type: application/taxii+json;version=2.1
X-TAXII-Date-Added-First: 2018-05-12T03:05:09.15811Z
X-TAXII-Date-Added-Last: 2018-05-12T03:05:17.275544Z

{
  "objects": [
    {
      "type": "relationship",
      ...
    }
  ]
}
```