Vertex

Synapse Bootcamp - Module 9

Pivoting and Traversal in Storm - Answer Key

Answer Key

Pivoting

Exercise 1 Answer

Objectives:
- **Write and execute basic Storm queries using pivot operations.**
- **Use the special "pivot to tags" operation.**
- **Leverage the 'uniq' Storm command to deduplicate results.**

Question 1: How can you **add** to the Storm query in order to **pivot** to the associated DNS A (**`inet:dns:a`**) records? How many DNS A records are returned?

- Add a **pivot** operation to your existing query:

  Explicit syntax:

  ```
  inet:ipv4=173.254.222.138 -> inet:dns:a:ipv4
  ```

  Implicit syntax:

  ```
  inet:ipv4=173.254.222.138 -> inet:dns:a
  ```

- There are **14** **`inet:dns:a`** nodes returned by the query:

Question 2: How can you **add** to your Storm query to **pivot** from the DNS A records to the associated FQDNs (**`inet:fqdn`** nodes)? How many FQDNs are returned?

- Add a **pivot** operation to your existing query:

  Explicit syntax:

  ```
  inet:ipv4=173.254.222.138 -> inet:dns:a:ipv4 :fqdn -> inet:fqdn
  ```

  Implicit syntax:

  ```
  inet:ipv4=173.254.222.138 -> inet:dns:a -> inet:fqdn
  ```

- There are **14** **`inet:fqdn`** nodes returned by the query:

Question 3: How can you **add** to your Storm query to **pivot to tags** and view the **`syn:tag`** nodes for the tags on the FQDNs?

- Add another **pivot** operation to your existing query. In order to **pivot to tags,** use the special hashtag (**`#`**) character as the target of the pivot:

  Explicit syntax:

  ```
  inet:ipv4=173.254.222.138 -> inet:dns:a:ipv4 :fqdn -> inet:fqdn
    -> #
  ```

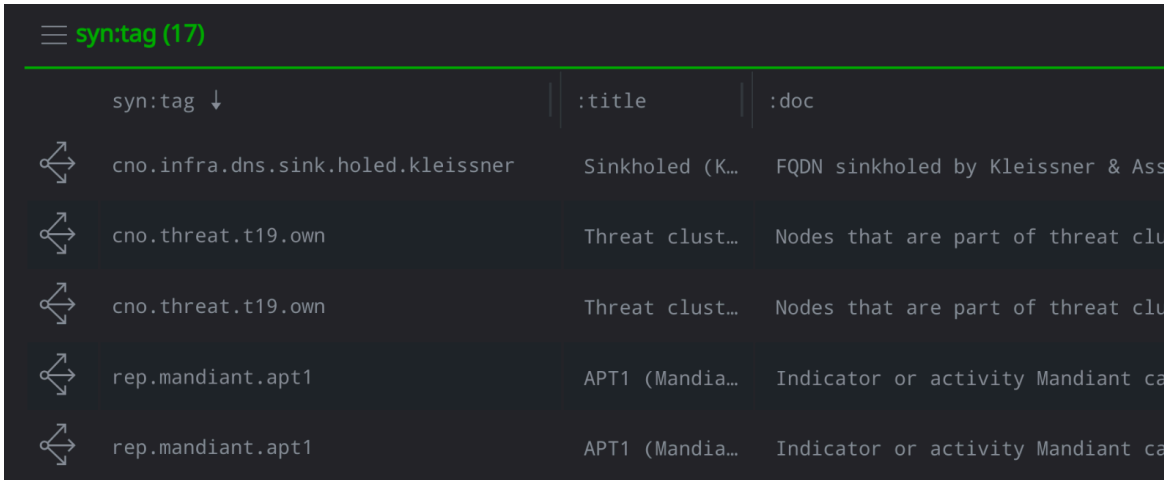
Implicit syntax:

```
inet:ipv4=173.254.222.138 -> inet:dns:a -> inet:fqdn -> #
```

> Because "pivot to tags" is a special pivot operation (it pivots specifically from **tags** on nodes to **syn:tag** nodes), there is no "explicit" vs "implicit" syntax for this operation; it is the same in both cases.

Question 4: How many **`syn:tag`** nodes are returned? Why are there duplicates?

- The query returns **17** **`syn:tag`** nodes:



  There are **duplicate** results because many of the **`inet:fqdn`** nodes (the **source** nodes for your pivot to tags operation) have the **same tag** applied to each node.

  For example: all 14 FQDNs have the tag **rep.mandiant.apt1**. When you pivot in Storm, Synapse returns one instance of the **`syn:tag`** node for **each** FQDN that has the tag. So there are 14 "copies" of the **rep.mandiant.apt1** tag in your results.

Question 5: What Storm command can you **add** to your Storm query to **deduplicate** ("unique") your results and only display one instance of each tag?

- Add the **`uniq`** command to the end of your query to "unique" the results:

  Explicit syntax:

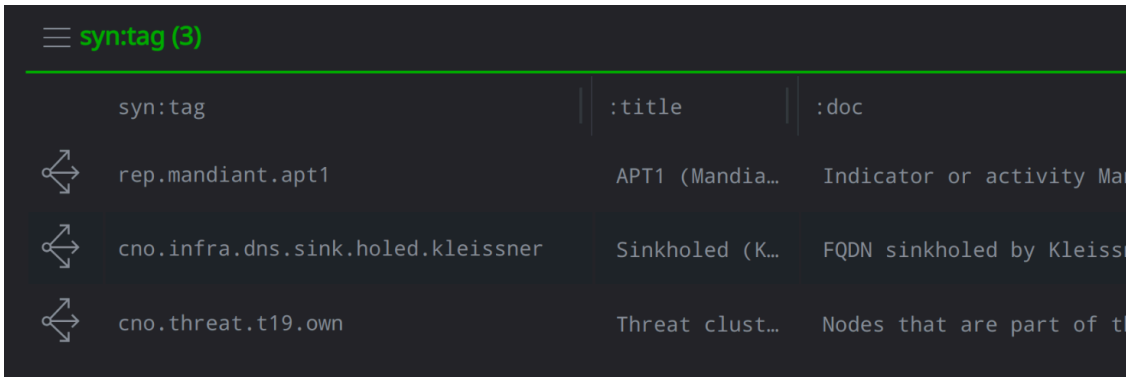
```
inet:ipv4=173.254.222.138 -> inet:dns:a:ipv4 :fqdn -> inet:fqdn
  -> # | uniq
```

Implicit syntax:

```
inet:ipv4=173.254.222.138 -> inet:dns:a -> inet:fqdn -> #
 | uniq
```

Recall that you use the pipe character (|) to switch from a Storm query to a Storm command and vice versa.

After running this query you should have only **three** `syn:tag` nodes in your results:



Running the **uniq** command is not affected by using either explicit or implicit syntax; the command can be added to either type of query.

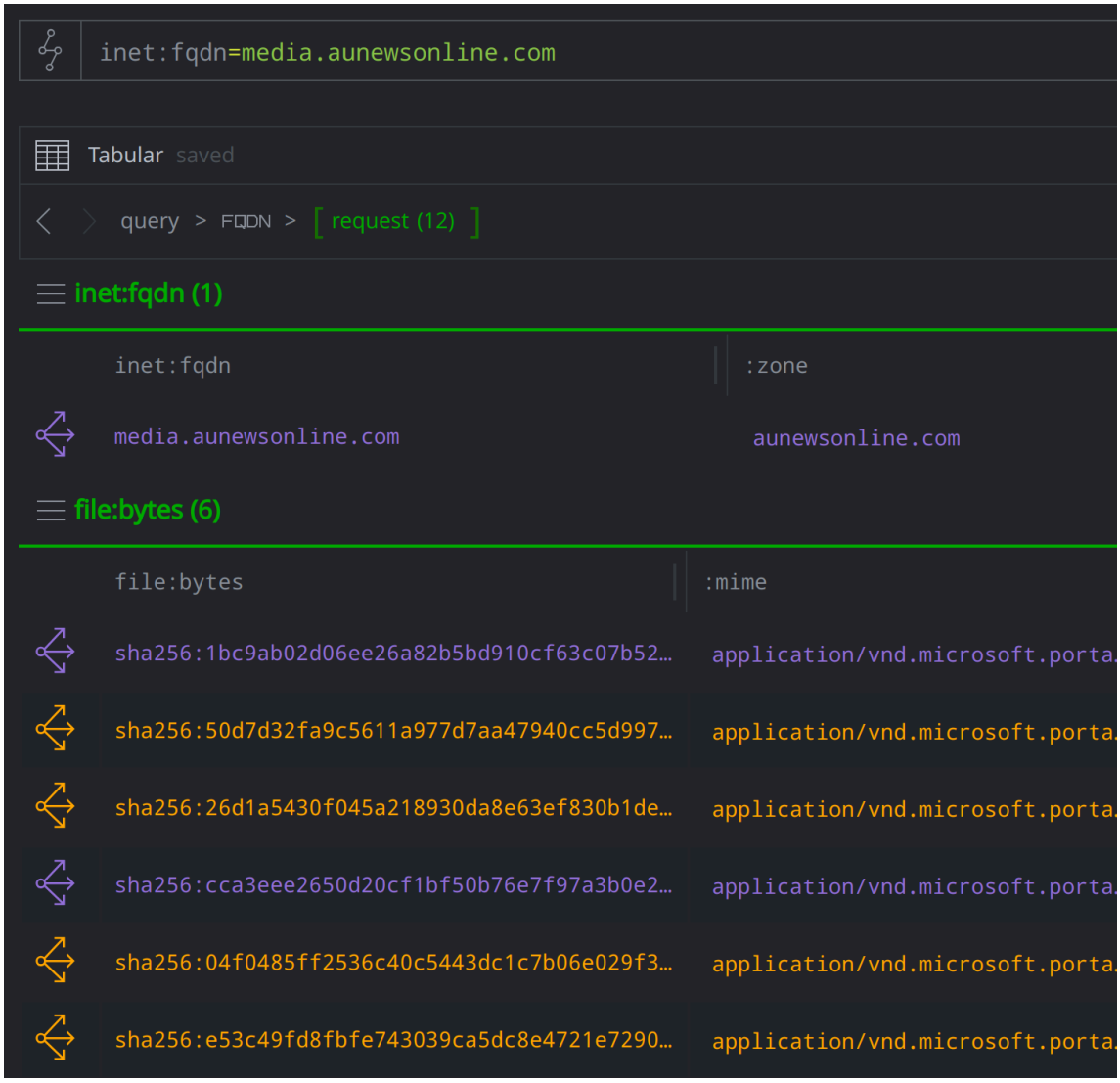
Exercise 2 Answer

Objectives:

- **Write and execute basic Storm queries using pivot operations.**
- **Compare and contrast use of the Synapse UI and use of Storm to examine data.**

Question 1: How many files perform DNS queries for the FQDN `media.aunewsonline.com`?

- There are **six** files that make DNS queries for that FQDN:



Question 2: How can you **add** to your Storm query to **pivot** to the associated DNS requests (**`inet:dns:request`** nodes)?

- Add a **pivot** operation to your query:

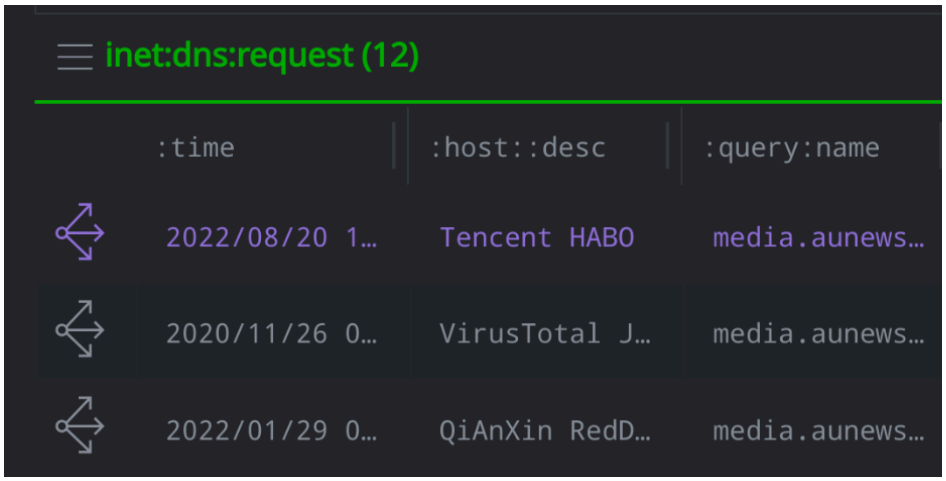
  Explicit syntax:

```
inet:fqdn=media.aunewsonline.com
 -> inet:dns:request:query:name:fqdn
```

  Implicit syntax:

```
inet:fqdn=media.aunewsonline.com -> inet:dns:request
```

You should see **12** `inet:dns:request` nodes:



Question 3: How can you **add** to your Storm query to **pivot** from the DNS requests to the files (`file:bytes`) that make the requests?

- Add another **pivot** operation to your query:

  Explicit syntax:

  ```
  inet:fqdn=media.aunewsonline.com
   -> inet:dns:request:query:name:fqdn :exe -> file:bytes
  ```

  Implicit syntax:

  ```
  inet:fqdn=media.aunewsonline.com -> inet:dns:request
   -> file:bytes
  ```

  You should see **24** `file:bytes` nodes:

Question 4: Do you have duplicate results? If so, how can you **add** to your query to remove deduplicate ("unique") the results?

- Add the **uniq** command to the end of your query to "unique" the results:

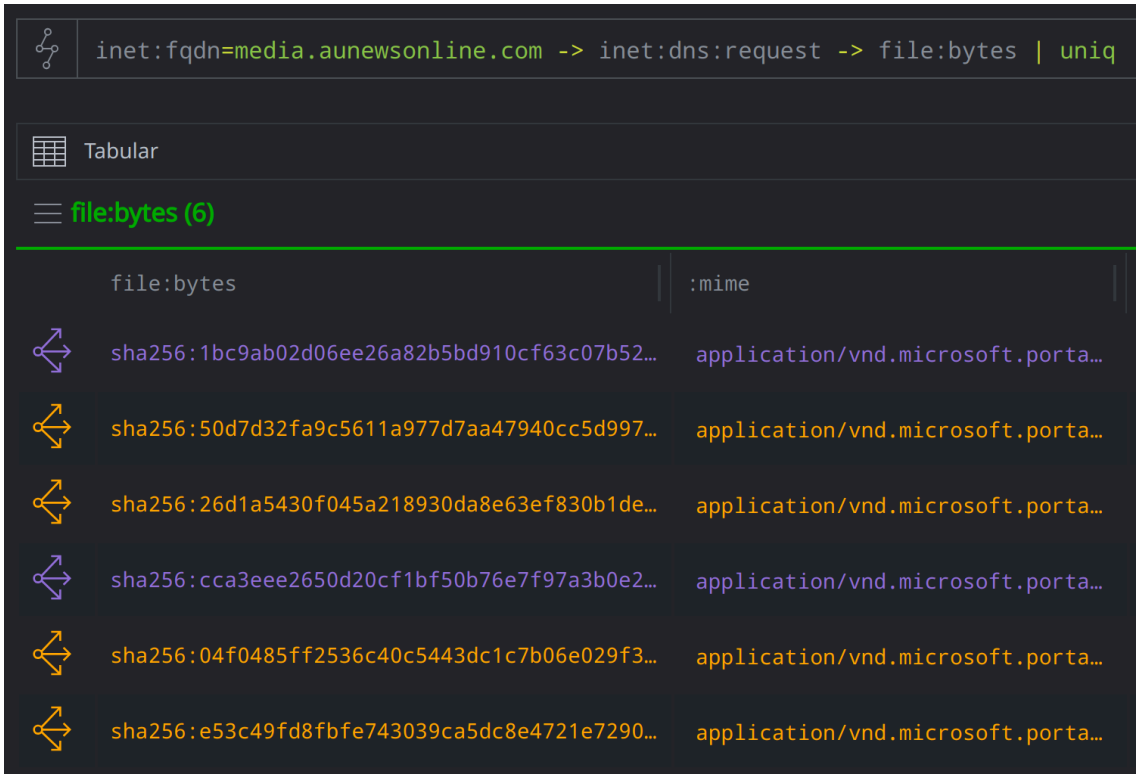
  Explicit syntax:

```
inet:fqdn=media.aunewsonline.com
 -> inet:dns:request:query:name:fqdn :exe -> file:bytes | uniq
```

  Implicit syntax:

```
inet:fqdn=media.aunewsonline.com -> inet:dns:request
  -> file:bytes | uniq
```

You should have **six** files - the same answer you received using the **Explore** button:



Edge Traversal

Exercise 3 Answer

Objective:
- **Write and execute basic Storm queries using edge traversal operations.**

Question 1: How can you **add** to your query to **traverse any light edges** and find **any** nodes that "point to" the FQDN?

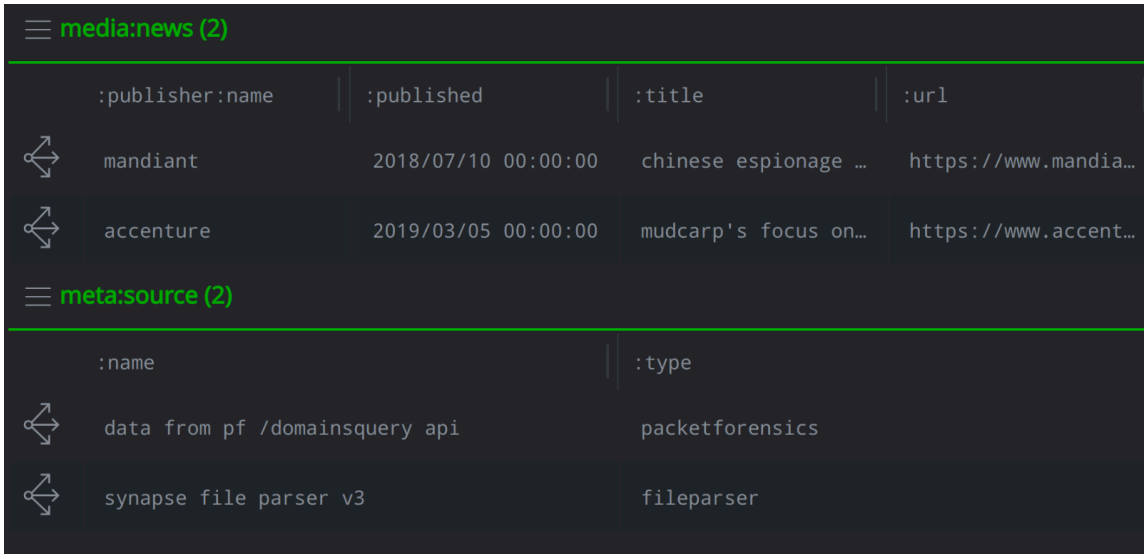
- To find any nodes that "point to" our FQDN using a light edge, we can use a **wildcard edge traversal** operation:

```
inet:fqdn=chemscalere.com <(*)- *
```

  Note that this syntax uses two wildcard characters (`*`) :
  - One in the edge traversal operator to indicate Synapse should traverse **any** light edge (`<(*)-`).
  - One representing the **target** forms for the operation (i.e., any / all forms).

Question 2: What kinds of nodes are linked to the FQDN using light edges?

- The FQDN is linked to:
  - Articles (**`media:news`** nodes)
  - Data sources (**`meta:source`** nodes)



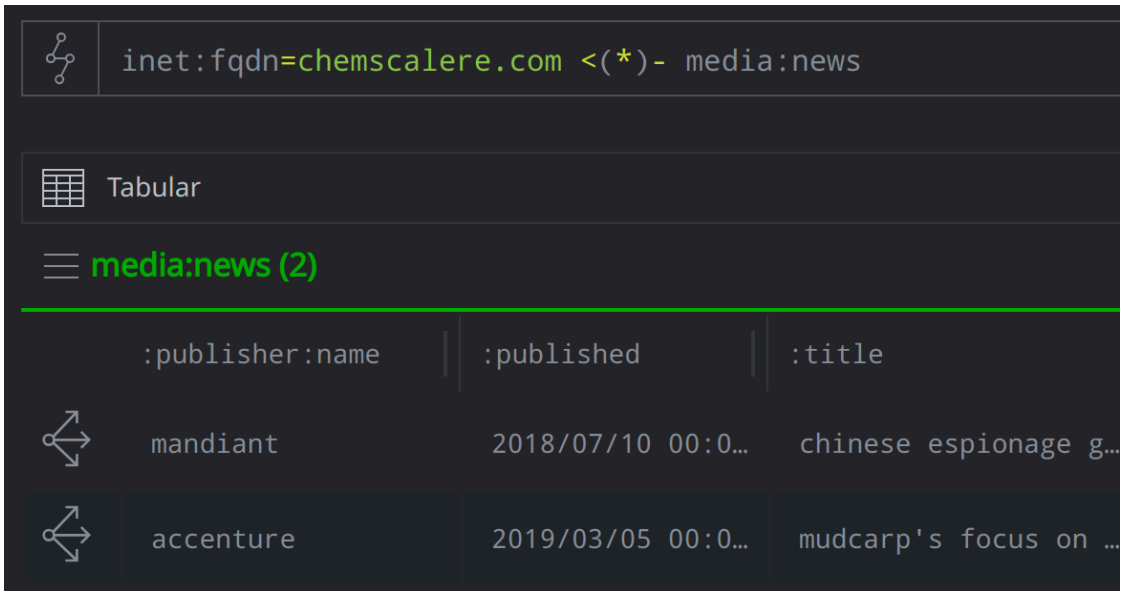
> **Note:** This answer is based on the baseline Synapse demo instance. Your answer may vary depending on any data that has been added to your demo instance so far in this course.

Question 3: How can you **modify** your Storm query to return **only** articles (**`media:news`** nodes) that include the FQDN?

- To return only the **`media:news`** nodes, we can change our target from a wildcard to only the nodes we want to see:

```
inet:fqdn=chemscalere.com <(*)- media:news
```

  We are still traversing "any" edge using the edge wildcard (**`<(*)-`**). But we are telling Synapse that the "target" nodes are articles (**`media:news`** nodes):



The following queries will **also** work:

- Tell Synapse you want to see **any** target nodes connected by a **refs** edge:

```
inet:fqdn=chemscalere.com <(refs)- *
```

  meta:source nodes are connected by a **seen** edge, so the above query will only show us the **media:news** nodes.

- Tell Synapse you want to see the **media:news** nodes connected by **refs** edges:

```
inet:fqdn=chemscalere.com <(refs)- media:news
```

Using Synapse with Large Data Sets

Exercise 4 Answer

Objectives:
- **Write and execute basic Storm queries using pivot operations for large data sets.**
- **Compare and contrast using the Synapse UI and using Storm to examine data.**

Question 1: What happens? Are you able to use the **Explore** button, or do you get stuck? (That is, do you get tired of loading results and wondering when Synapse will finish?)

- **You get "stuck".**

  Synapse will load results as long as you allow it to continue. But your browser will become less and less responsive as you load more and more results. Synapse can handle large data sets - but your browser cannot.

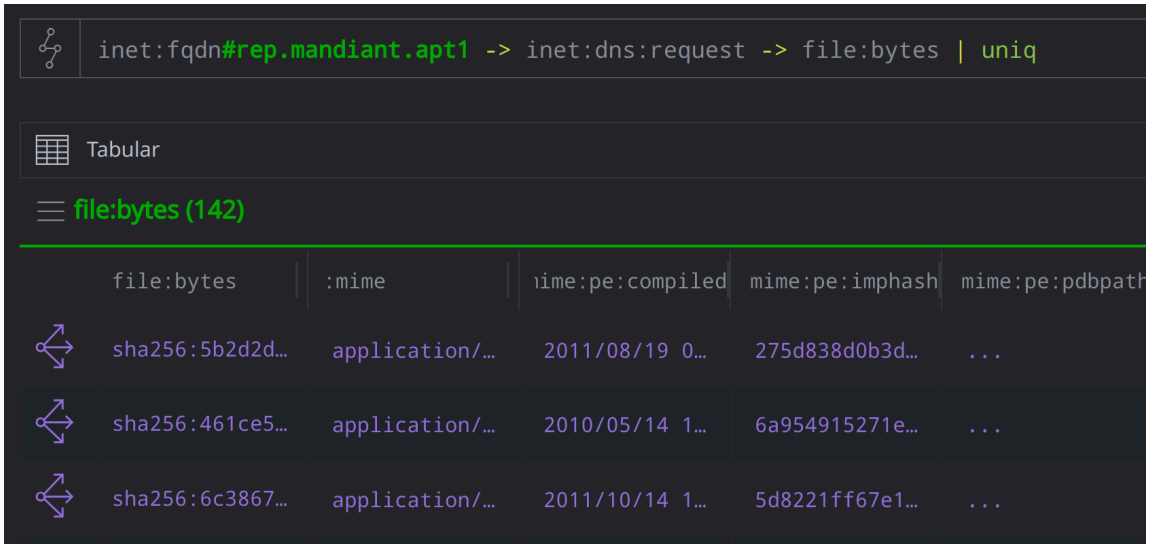
  To answer our question, we're starting with **all** of the APT1 FQDNs - 2,073 of them.

  When we **Explore**, we ask Synapse to take **all** 2,000+ nodes and show us **all** the things those nodes are connected to (over 54,000 nodes!) - even though we really only care about the DNS requests.

Question 2: What happens? How many files are returned?

- When we use Storm to ask the **exact** question we want to answer, we get a response almost immediately!

  There are **142** files that communicate with any of the 2,073 APT1 FQDNs:



  When we ask Synapse to show us **precisely** the data we're looking for, Synapse is much more efficient - there is no need to select, display, and navigate through data we don't care about to (eventually) get our answer.
