

Synapse Bootcamp - Module 11

Building Queries in Storm - Exercises

Building Queries in Storm - Exercises	1
Objectives	1
Exercises	3
Storm Commands	3
Exercise 1	3
Building a Storm Query	5
Exercise 2	5
Working with Bookmarks	10
Exercise 3	10
Working with Storm Editor	15
Exercise 4	15

Objectives

In these exercises you will learn:

- How to use common Storm commands to aid analysis
- How Storm's operating concepts apply to real world queries
- How to build a Storm query "as you go" to explore data and follow an analytical path through Synapse's data
- How to bookmark a query and add it to your Favorites
- How to use the features of the Storm Editor tool to create and test a query

Note: We are constantly updating Synapse and its Power-Ups! We do our best to make sure our course documents (slides, exercises, and answer keys) are up-to-date. However, you may notice small differences (such as between a screen capture in the documents and the appearance of your current instance of Synapse).

If something is unclear or if you identify an error, please reach out to us so we can assist!

Exercises

- All exercises use the **Research Tool** with the **Storm Mode Selector** set to **Storm mode**.
- Some example queries may wrap due to length.

The **Storm Quick Reference** guides (included with the supplemental materials provided for this course) may be helpful for this (and future) exercises.

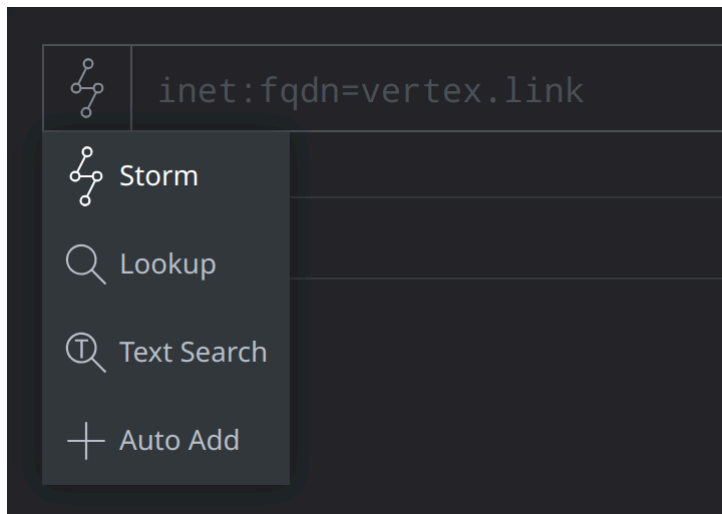
The online [Storm Reference](#) includes detailed documentation and examples for all things Storm!

Storm Commands

Exercise 1

Objective:

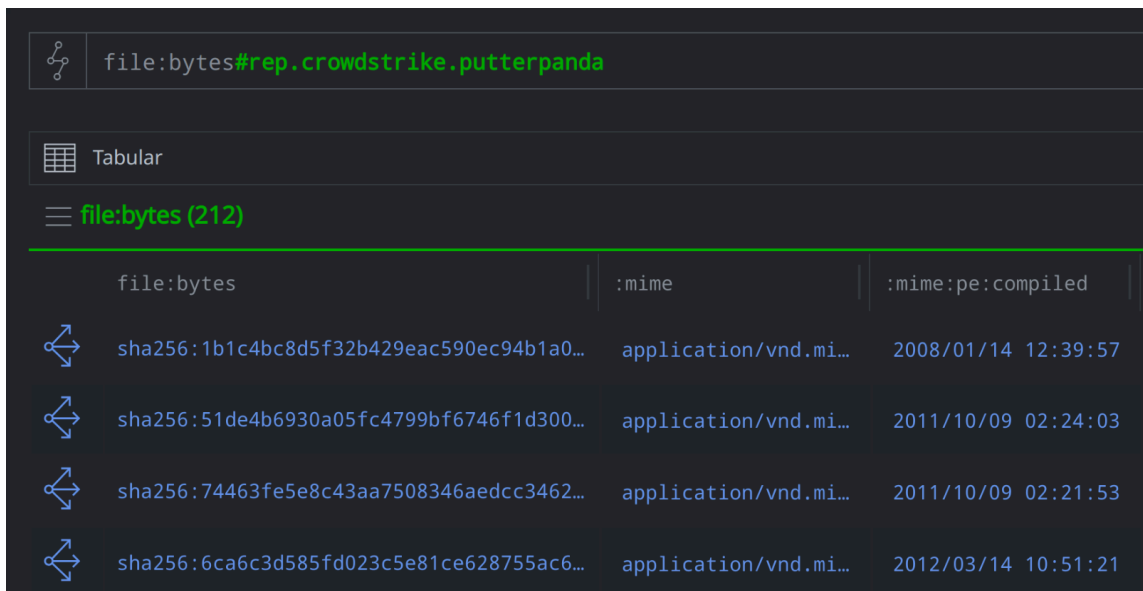
- Use common Storm commands to help answer analytical questions.
- In the **Research Tool**, ensure your **Storm Query Bar** is in **Storm mode**:



You are viewing the files (**file:bytes**) associated with the Putter Panda threat group.

- Enter the following in the **Storm Query Bar** and press **Enter** to **lift** the nodes:

```
file:bytes#rep.crowdstrike.putterpanda
```



file:bytes	:mime	:mime:pe:compiled
sha256:1b1c4bc8d5f32b429eac590ec94b1a0...	application/vnd.mi...	2008/01/14 12:39:57
sha256:51de4b6930a05fc4799bf6746f1d300...	application/vnd.mi...	2011/10/09 02:24:03
sha256:74463fe5e8c43aa7508346aedcc3462...	application/vnd.mi...	2011/10/09 02:21:53
sha256:6ca6c3d585fd023c5e81ce628755ac6...	application/vnd.mi...	2012/03/14 10:51:21

Question 1: What **Storm command** can you add to the end of your query to find a file with the **earliest compile (:mime:pe:compiled)** time?

Hint: Remember to use the pipe character (|) in Storm to switch between query mode and command mode, and to separate individual commands.

Question 2: What is the earliest compile date/time?

You want to know how many **total** indicators in Synapse are associated with Putter Panda (**rep.crowdstrike.putterpanda**).

Question 3: What Storm query / command can you use to determine the total number of indicators (without displaying the nodes)?

Question 4: How many indicators are there?

Building a Storm Query

Exercise 2

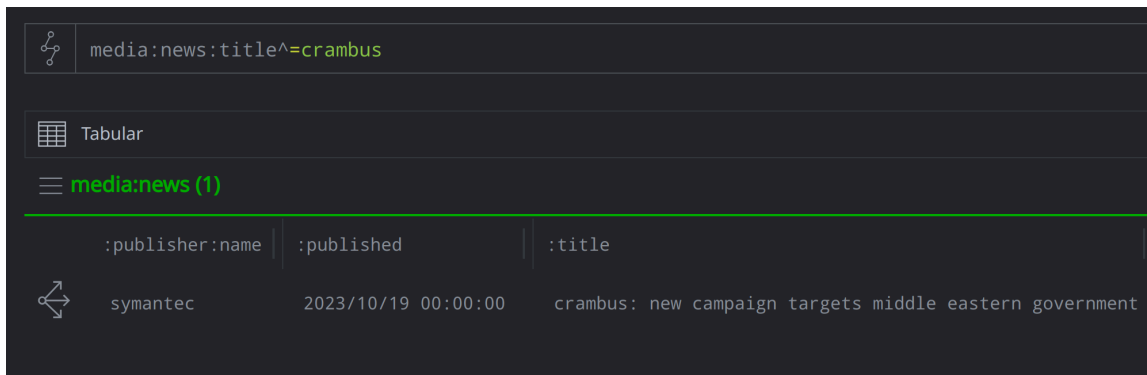
Objective:

- Understand how to build a Storm query step-by-step as you explore data and conduct analysis.

You are reviewing a Symantec blog on activity by the **Crambus** threat group. The blog has been ingested into Synapse as a **media:news** node, which is the starting point for your research.

- Enter the following in the **Storm Query Bar** and press **Enter** to **lift** the **media:news** node:

```
media:news:title^=crambus
```

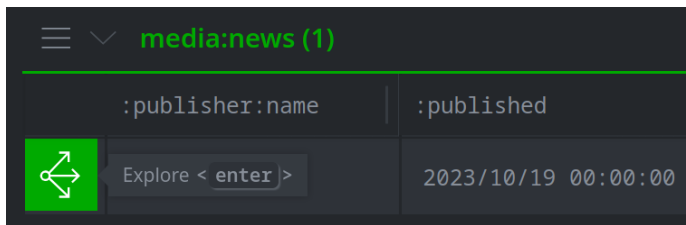


The screenshot shows a dark-themed interface with a search bar containing the query `media:news:title^=crambus`. Below the search bar, there is a "Tabular" view option. The results are displayed as a table with one row of data. The table has columns for publisher name, published date, and title.

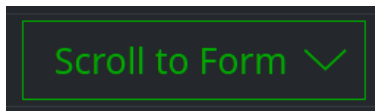
:publisher:name	:published	:title
symantec	2023/10/19 00:00:00	crambus: new campaign targets middle eastern government

You want to view the data associated with (referenced by) the article to get an idea of the content. We'll use the **Explore** button to get a quick overview.

- Click the **Explore** button next to the **media:news** node to navigate to the adjacent nodes:



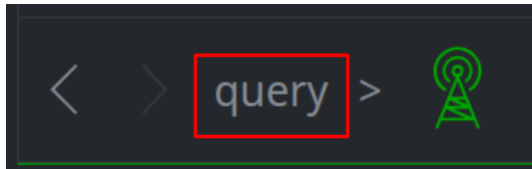
- **Browse** your results (use **Scroll to Form** if necessary):



Question 1: What kinds of nodes are connected to the article?

You decide you want to examine the IPv4 addresses (**inet:ipv4** nodes) from the article.

- In your **Breadcrumbs**, click **query** to return to the **media:news** node:



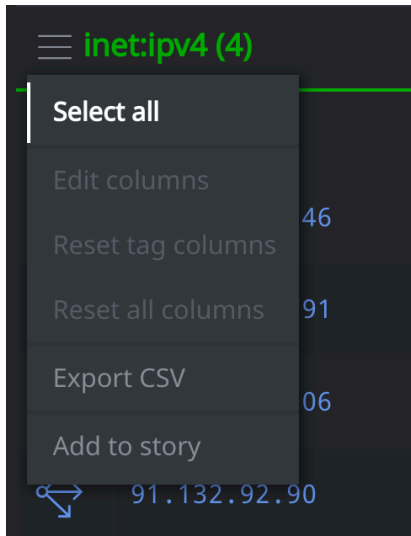
Question 2: What Storm operation can you add to your original query to **only** view the **inet:ipv4** nodes from the article?

The IPv4 addresses (**inet:ipv4**) referenced by the article include indicators of compromise (IOCs) reported by Symantec, but also some non-routable addresses. You want to **only** examine the IPv4s reported as IOCs.

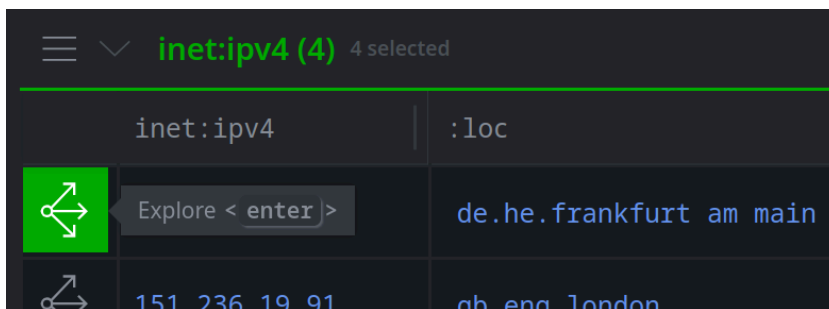
Question 3: What Storm operation can you add to your query to **only** view the IOCs reported by Symantec (**rep.symantec**)?

Your results now consist of only four IPv4 addresses, but you're not sure where to go from here. Let's see what else might be nearby.

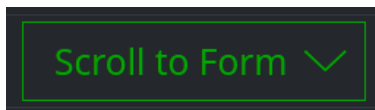
- Click the hamburger menu to the left of the **inet:ipv4** table header and choose **Select All**:



- Click the **Explore** button next to any selected IPv4 address to display adjacent nodes:

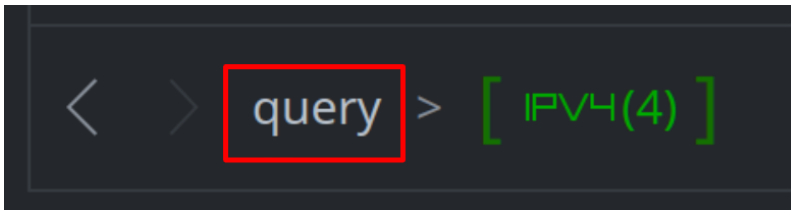


- Click the **Scroll to Form** button to see the kinds of nodes in your results:



The results include **inet:server** nodes, which show the open ports, protocols, and services that were seen on these IPv4 addresses. You want to investigate these servers.

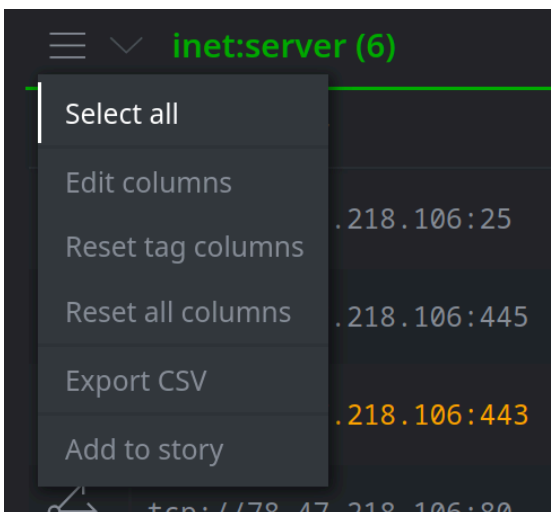
- In your **Breadcrumbs**, click **query** to return to the **inet:ipv4** addresses:



Question 4: What Storm operation can you add to your query to **pivot** to the **inet:server** nodes?

Let's continue and see what is connected to our **inet:server** nodes.

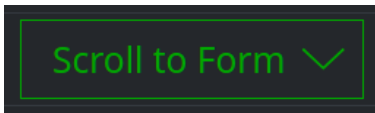
- Click the hamburger menu to the left of the **inet:server** table header and choose **Select All**:



- Click the **Explore** button next to any selected server to display adjacent nodes:

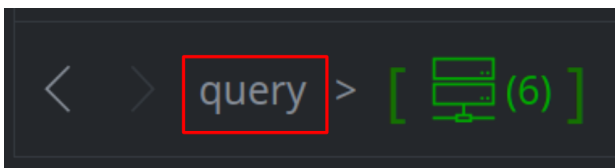


- Click the **Scroll to Form** button to see the kinds of nodes in your results:



Your results include an **inet:tls:servercert** node. This indicates that a TLS or SSL certificate (**crypto:x509:cert**) was hosted on at least one of the servers (**inet:server**). You want to examine the certificate file.

- In your **Breadcrumbs**, click **query** to return to the **inet:server** nodes:

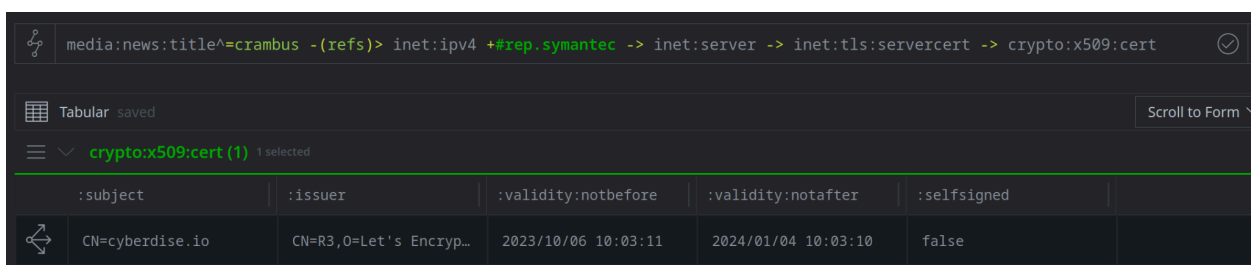


Question 5: What Storm operation can you add to your query to **pivot** to the **inet:tls:servercert** node?

The **inet:tls:servercert** node represents a certificate hosted on a server. You want to see more details about the certificate itself (the **crypto:x509:cert** node). (While you could use the **Explore** button to determine the next operation, we'll save a bit of time.)

- Add the following **pivot** operation to your existing query to pivot from the **inet:tls:servercert** node to the **crypto:x509:cert** node:

```
media:news:title^=crambus -(refs)> inet:ipv4 +#rep.symantec
-> inet:server -> inet:tls:servercert -> crypto:x509:cert
```



The screenshot shows a query interface with the following query: `media:news:title^=crambus -(refs)> inet:ipv4 +#rep.symantec -> inet:server -> inet:tls:servercert -> crypto:x509:cert`. The results are displayed in a tabular view with the following columns and data:

:subject	:issuer	:validity:notbefore	:validity:notafter	:selfsigned
CN=cyberdisse.io	CN=R3,O=Let's Encryp...	2023/10/06 10:03:11	2024/01/04 10:03:10	false

A `crypto:x509:cert` node represents the **metadata** associated with a certificate file - information like the certificate's issuer and validity dates. You have some information about the certificate already, but want to try and further enrich the certificate using the Synapse-Censys Power-Up (specifically, the `censys.certs.enrich` command).

While you could run this command as a Node Action, you want to show off your Storm skills!

Question 6: What Storm operation can you **add** to this query to run the `censys.certs.enrich` command?

Working with Bookmarks

Exercise 3

Objective:

- Save a useful query as a bookmark.
- Add the bookmark to your Favorites for easy access.

The following Storm query helps us hunt for "unknown" malware:

```
inet:fqdn#rep inet:fqdn#cno | uniq | :zone -> inet:fqdn:zone | uniq  
| -> inet:dns:request -> file:bytes -#cno | uniq
```

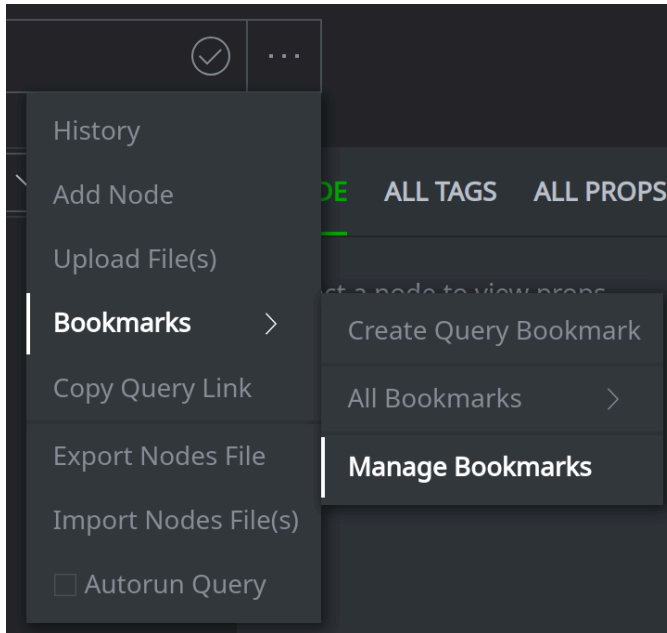
This query:

- takes "known bad" FQDNs (tagged `rep` for third-party reporting or `cno` for Vertex assessments) and their subdomains;
- looks for files that make DNS requests to any of those domains;
- filters out files that Vertex has already flagged "known bad" (tagged `cno`); and
- shows us the results: that is, files that query malicious FQDNs but are **not** currently tagged as malicious by Vertex.

Tip: If you are interested in the **step-by-step** breakdown of the operations in the above Storm query, the details are in the **Answer key**.

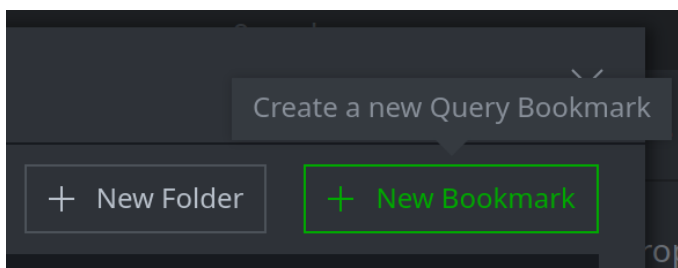
Because new data is constantly being added to Synapse, you want to save this query and run it periodically to hunt for new malware.

- In the **Research Tool**, click the **Storm Query Bar Menu** and choose **Bookmarks > Manage Bookmarks**:



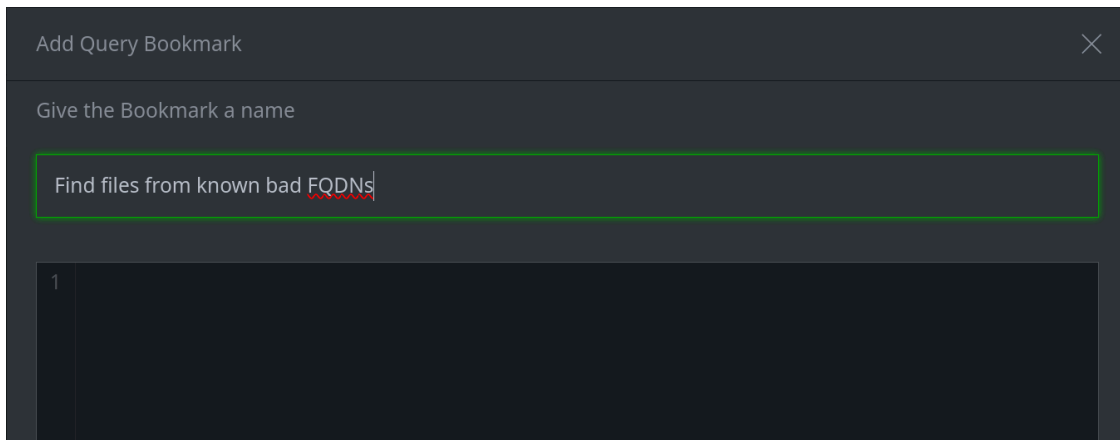
Note: The **Manage Bookmarks** option allows you to manually create a new bookmark. The **Create Query Bookmark** will create a Bookmark based on the current query in the Storm query bar.

- In the **Query Bookmarks** dialog, click the **+ New Bookmark** button to create a new bookmark:



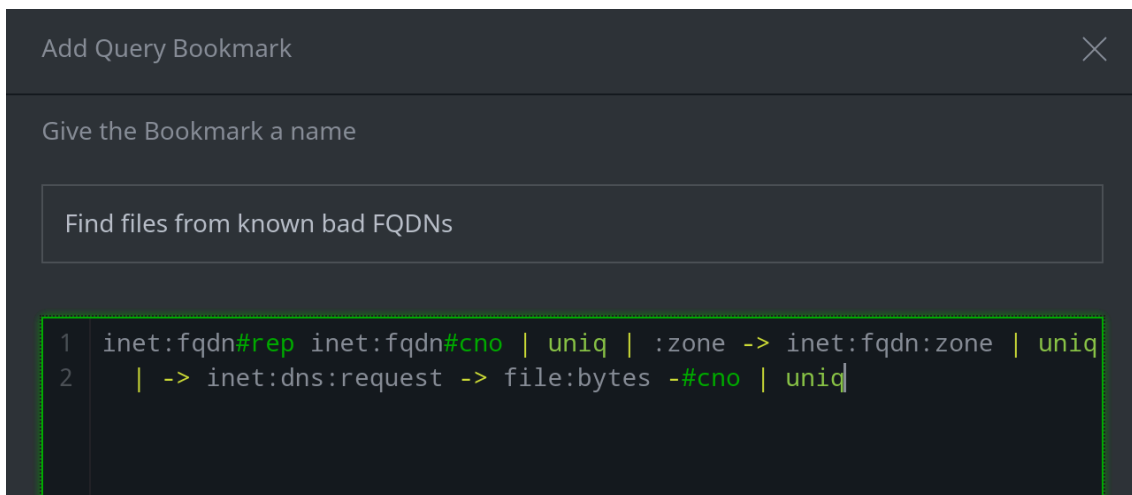
- In the **Add Query Bookmark** dialog, enter the following in the *Bookmark name* field:

Find files from known bad FQDNs



- Paste the "hunt" query into the **Storm Editor** window:

```
inet:fqdn#rep inet:fqdn#cno | uniq | :zone -> inet:fqdn:zone |  
uniq | -> inet:dns:request -> file:bytes -#cno | uniq
```

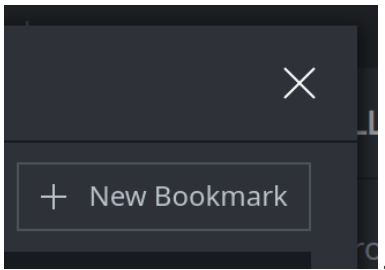


Tip: You can optionally insert a line break so you do not need to scroll to view the full query.

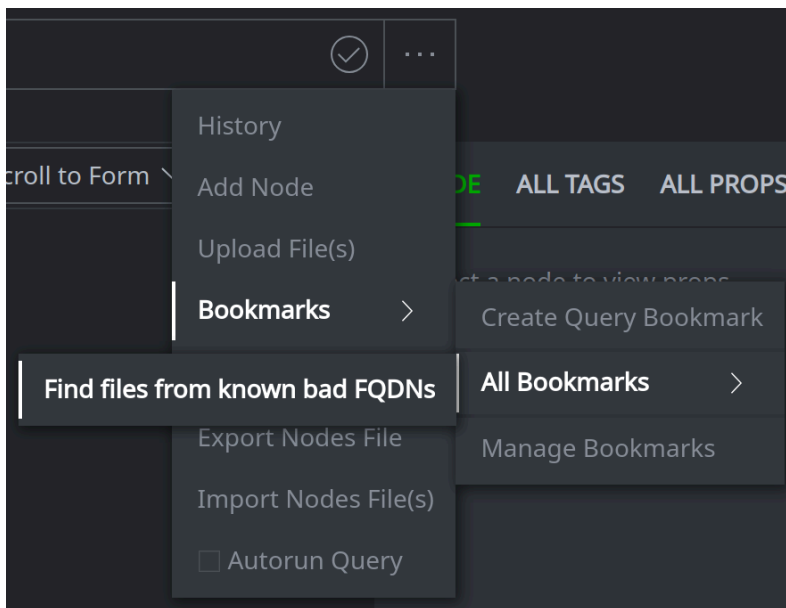
- Click the **Save Bookmark** button to create the bookmark:



- Click the **X** in the upper right to **close** the **Query Bookmarks** dialog:



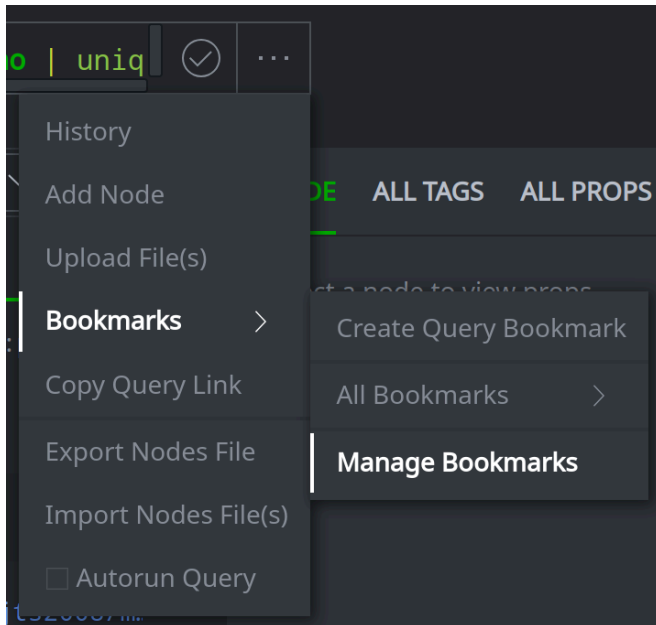
- Click the **Storm Query Bar Menu** and choose **Bookmarks > All Bookmarks > Find files from known bad FQDNs** to load and run your query:



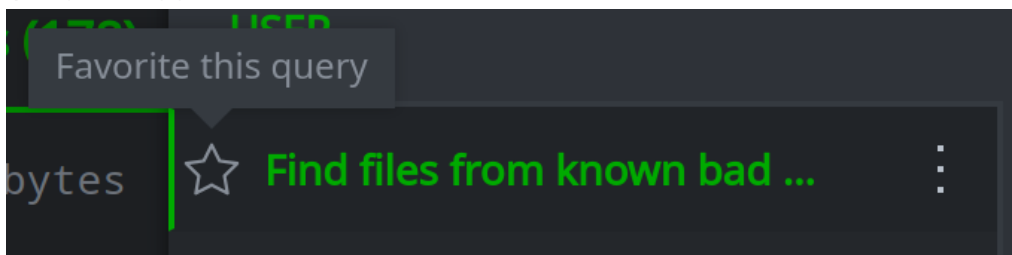
Question 1: How many files did your query find?

This is a pretty cool query so we want to add it to our Favorites.

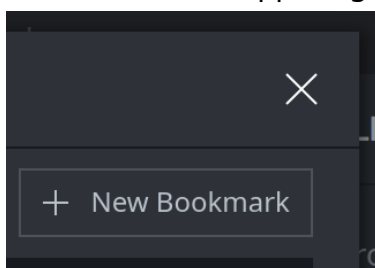
- Click the **Storm Query Bar Menu** and choose **Bookmarks > Manage Bookmarks**:



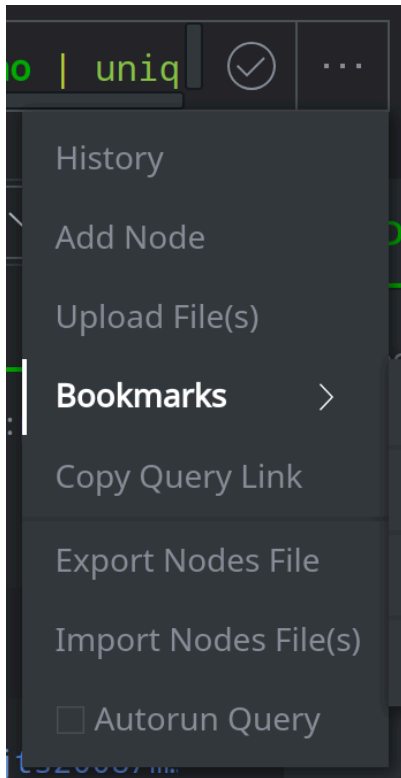
- In the **Query Bookmarks** dialog, in the list view, click the **star** icon next to your query to toggle the star **on**:



- Click the **X** in the upper right to **close** the dialog:



- Click the **Storm Query Bar Menu** and choose **Bookmarks** again:



Question 2: Where is your bookmarked query located now?

Working with Storm Editor

Exercise 4

Objective:

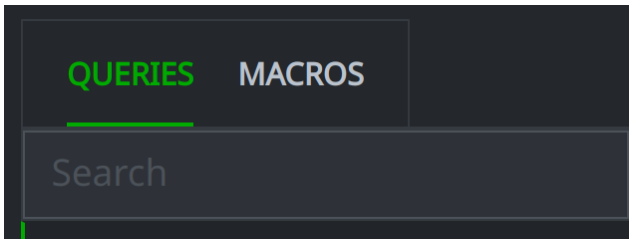
- Use Storm Editor to write and test a Storm query.
- Add comments to the query.

IF TIME ALLOWS:

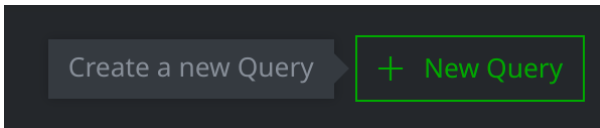
Practice using Storm Editor with the query from the previous exercise:

```
inet:fqdn#rep inet:fqdn#cno | uniq | :zone -> inet:fqdn:zone | uniq  
| -> inet:dns:request -> file:bytes -#cno | uniq
```

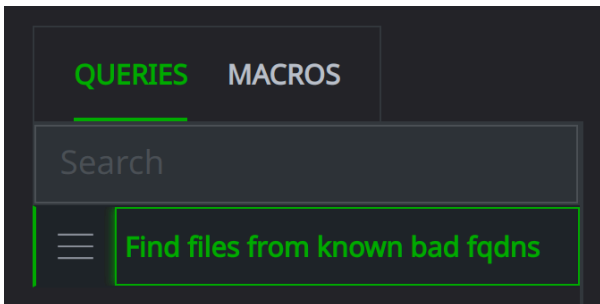
- In the **Storm Editor Tool**, click the **Queries** tab:



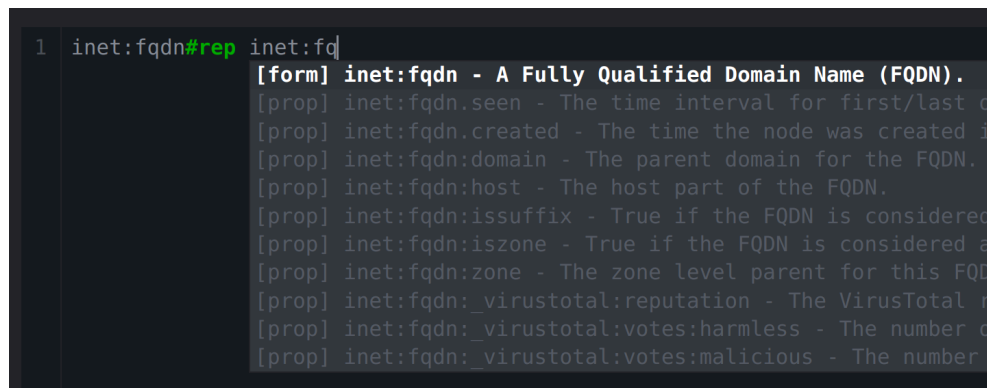
- Click the **+ New Query** button in the upper right to create a new query:



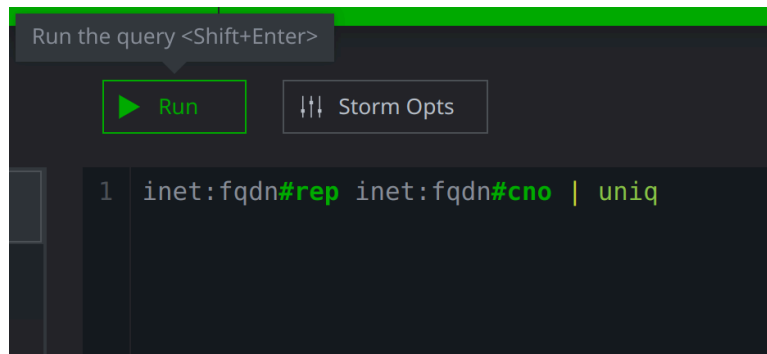
- Specify a name for the query:



- Use the query above to experiment with the features of the Storm Editor window. You may wish to:
 - Enter a portion of the query manually and test Synapse's **auto-complete** options:



- Build the query **step-by-step**, adding one operation at a time and using the **Run** button to test each step:



- Notice how the **syntax highlighting** changes as you type.
- Add **line breaks** for readability. (Synapse will execute the query the same way regardless of line breaks, as long as you do not place a line break in the middle of a single operation.)
- Add **comments** to the query. Storm supports:

- Single-line comments, using either double forward slashes (`//`) or "slash star" separators (`/*` and `*/`):

```
// This is a single line comment.
```

```
/* This is also a single line comment. */
```

- Multi-line comments using the "slash star" separators:

```
/*  
    This is a multi-line comment  
    that spans more than one line.  
*/
```

```
▶ Run Storm Opts
1 inet:fqdn#rep inet:fqdn#cno | uniq | :zone -> inet:fqdn:zone | uniq
2
3 /*
4 This is a comment!
5 */
6
7 | -> inet:dns:request -> file:bytes -#cno | uniq
8
```
