Synapse Bootcamp - Module 15

Static Malware Analysis - Exercises

Objectives

In these exercises you will:

- Use Power-Ups to ingest data useful for static analysis.
- Use the FileParser Power-Up to extract and view data from files.
- Use static data such as file metadata, multiscanner data, and code signing data to pivot through Synapse and hunt for potentially related samples.

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 Jump Start** (included with the supplemental materials provided for this course) includes sample Storm queries / pivots for some common analysis tasks and may be useful for this module.

Static Malware Analysis

Exercise 1

Objective:

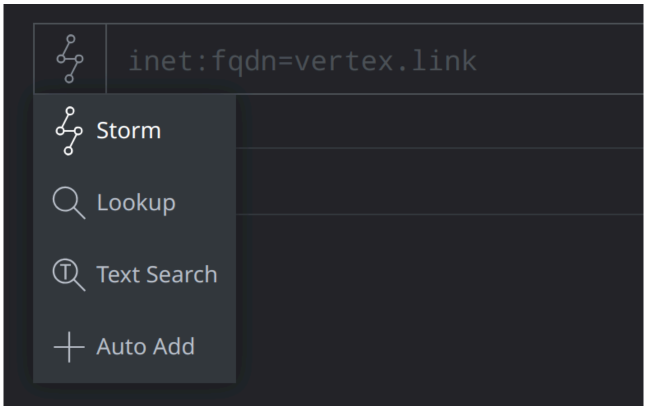
- **Use Power-Ups to enrich and research a suspicious file.**
- **Examine static data to gain insight into the file.**

Part 1

A customer has provided you with the SHA256 hash of a suspicious file and wants to know what you can tell them about it.

First we will see if we can download and parse a copy of the file.

- In the **Research Tool**, ensure your **Storm Query Bar** is in **Storm mode:**

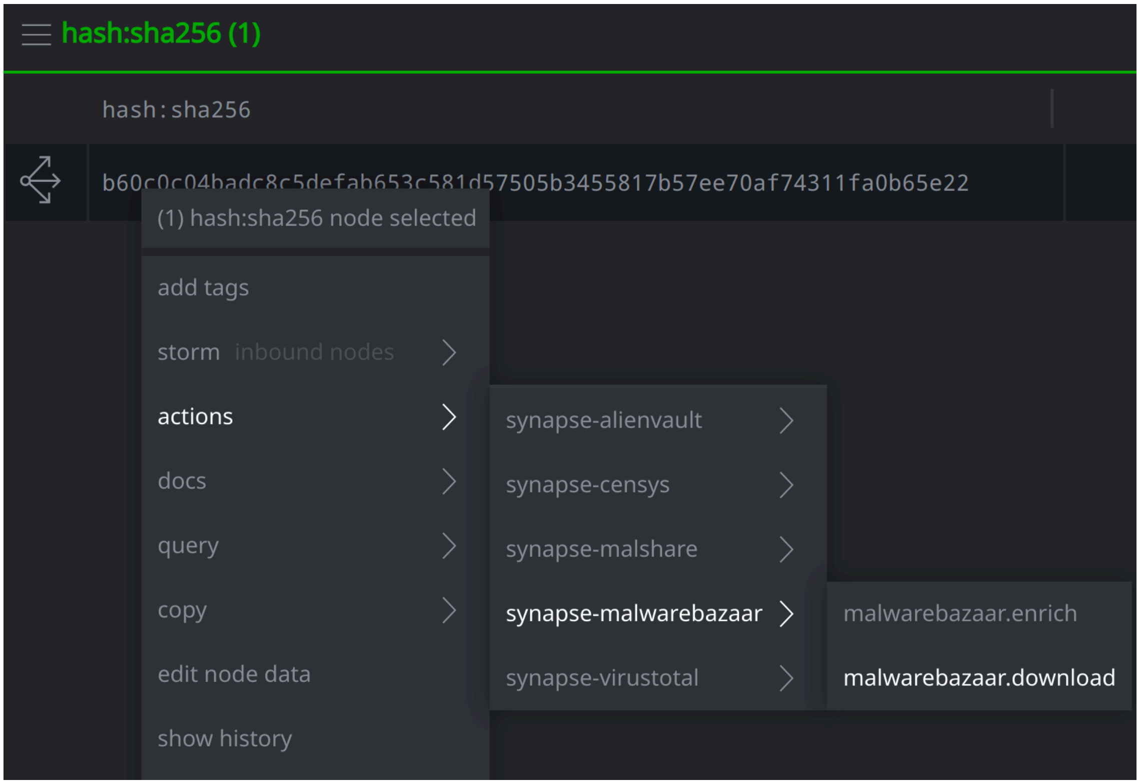


- Enter the following into your **Storm Query Bar** and press **Enter** to run the query:

```
[hash:sha256=b60c0c04badc8c5defab653c581d57505b3455817b57ee70af
74311fa0b65e22]
```

Note: The exercise PDFs may insert line breaks or spaces where values (such as the SHA256, above) are forced to wrap. If you copy the above into your Storm query bar and the query fails to run, you may need to manually remove the space / break.

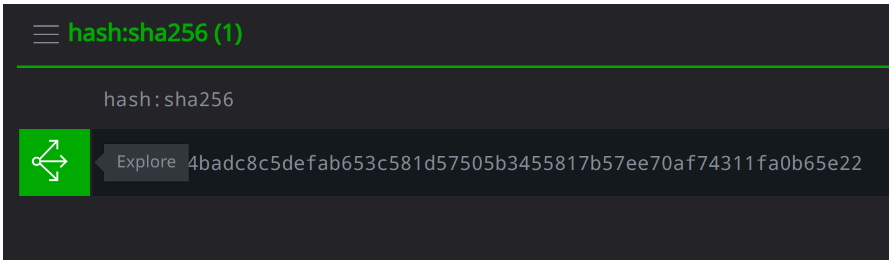
- In your **Results Panel,** select the `hash:sha256` node. Right-click the hash and select **actions > synapse-malwarebazaar > malwarebazaar.download** to try to download the associated file:


hash:sha256 (1)

hash:sha256
b60c0c04badc8c5defab653c581d57505b3455817b57ee70af74311fa0b65e22

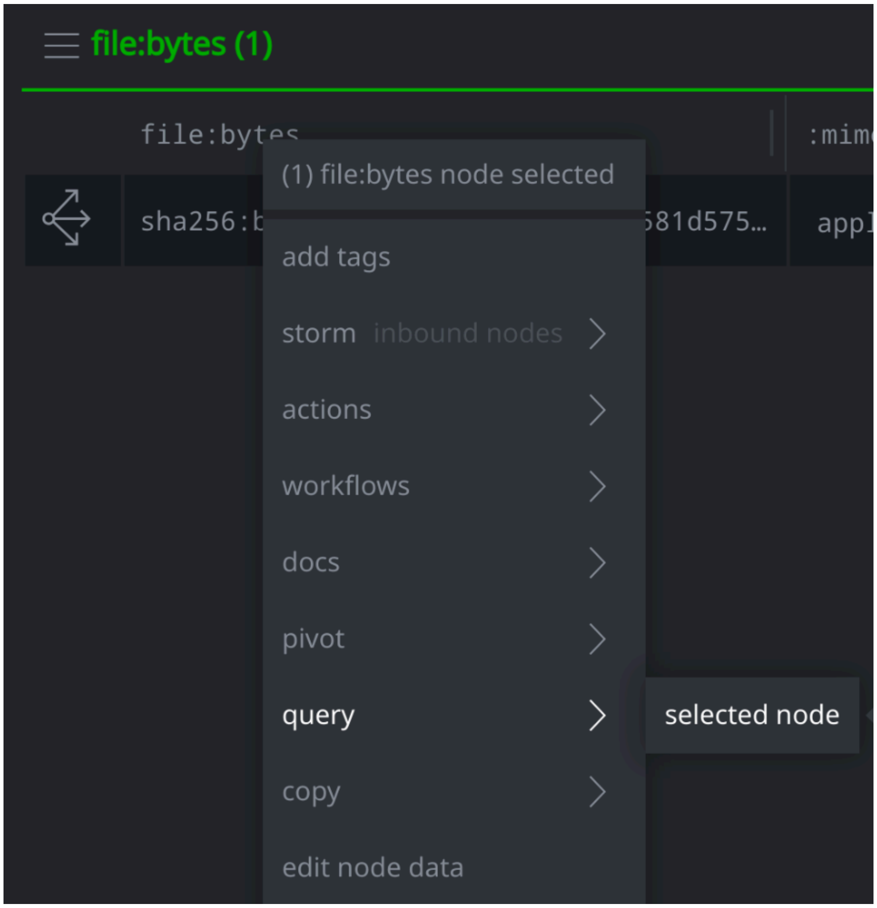
Question 1: Were you able to download the file? How can you tell?

- In your **Results Panel,** select the **`hash:sha256`** node. Click the **Explore** button next to the hash to pivot to the resulting **`file:bytes`** node:

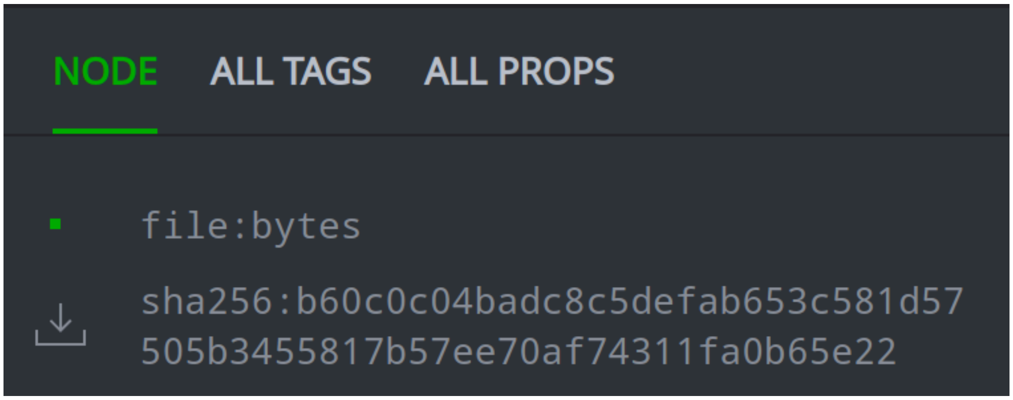


4badc8c5defab653c581d57505b3455817b57ee70af74311fa0b65e22

- Right-click the **`file:bytes`** node in your **Results Panel** and select **query > selected node** to run a new Storm query to lift the file (instead of our original **`hash:sha256`**):



- **Select** the **`file:bytes`** node and view its properties in the **Details Panel**:



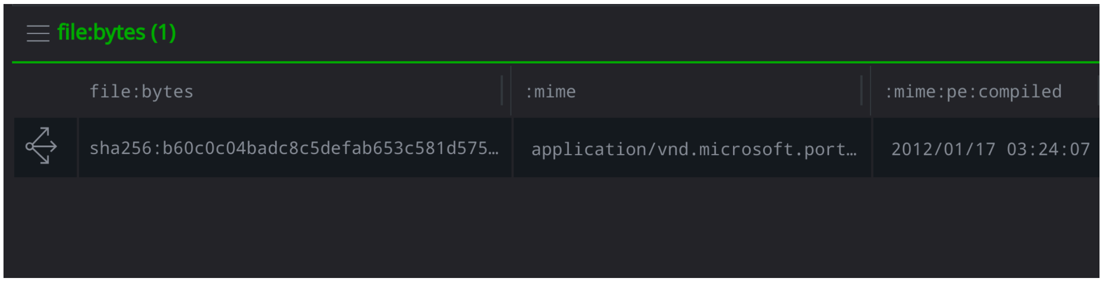
NODE ALL TAGS ALL PROPS

```
file:bytes
sha256:b60c0c04badc8c5defab653c581d57
505b3455817b57ee70af74311fa0b65e22
```

Question 2: What properties are set for the file?

PE files that are compiled in the same way or in the same environment may share properties, such as their import hash[1] value, compile time, or PDB path. We want to know if there are other files in Synapse that share some of these properties.

- In your **Results Panel,** select the **`file:bytes`** node

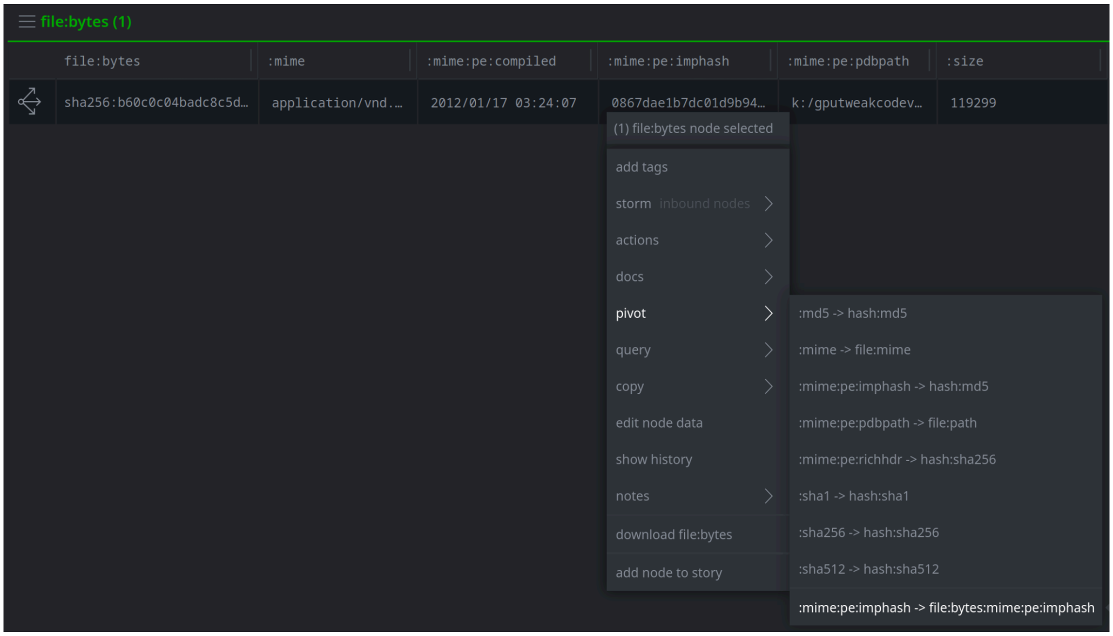


file:bytes (1)

file:bytes	:mime	:mime:pe:compiled
sha256:b60c0c04badc8c5defab653c581d575…	application/vnd.microsoft.port…	2012/01/17 03:24:07

[1] An **[import hash](#)** is an MD5 hash of the functions imported by a PE executable file. Files with the same import hash value use the same imported functions in the same order. This may help identify similar files (in this case, related malware samples) that were compiled from the same or similar source code.

- Right click the value in the **:mime:pe:imphash** column and select **pivot > :mime:pe:imphash > file:bytes:mime:pe:imphash** to lift all files in Synapse with this import hash value:



file:bytes (1)

file:bytes	:mime	:mime:pe:compiled	:mime:pe:imphash	:mime:pe:pdbpath	:size
sha256:b60c0c04badc8c5d…	application/vnd.…	2012/01/17 03:24:07	0867dae1b7dc01d9b94…	k:/gputweakcodev…	119299

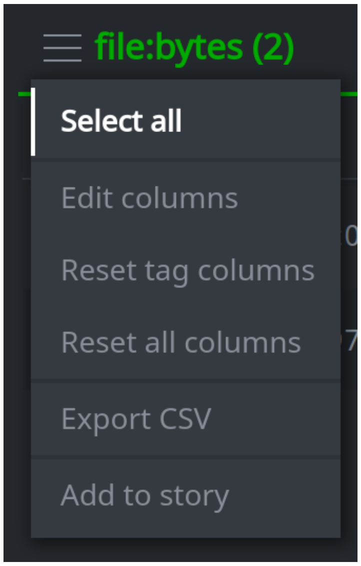
Question 3: Do any other files in Synapse share this import hash value?

Question 4: Do you notice any other similarities between the files?

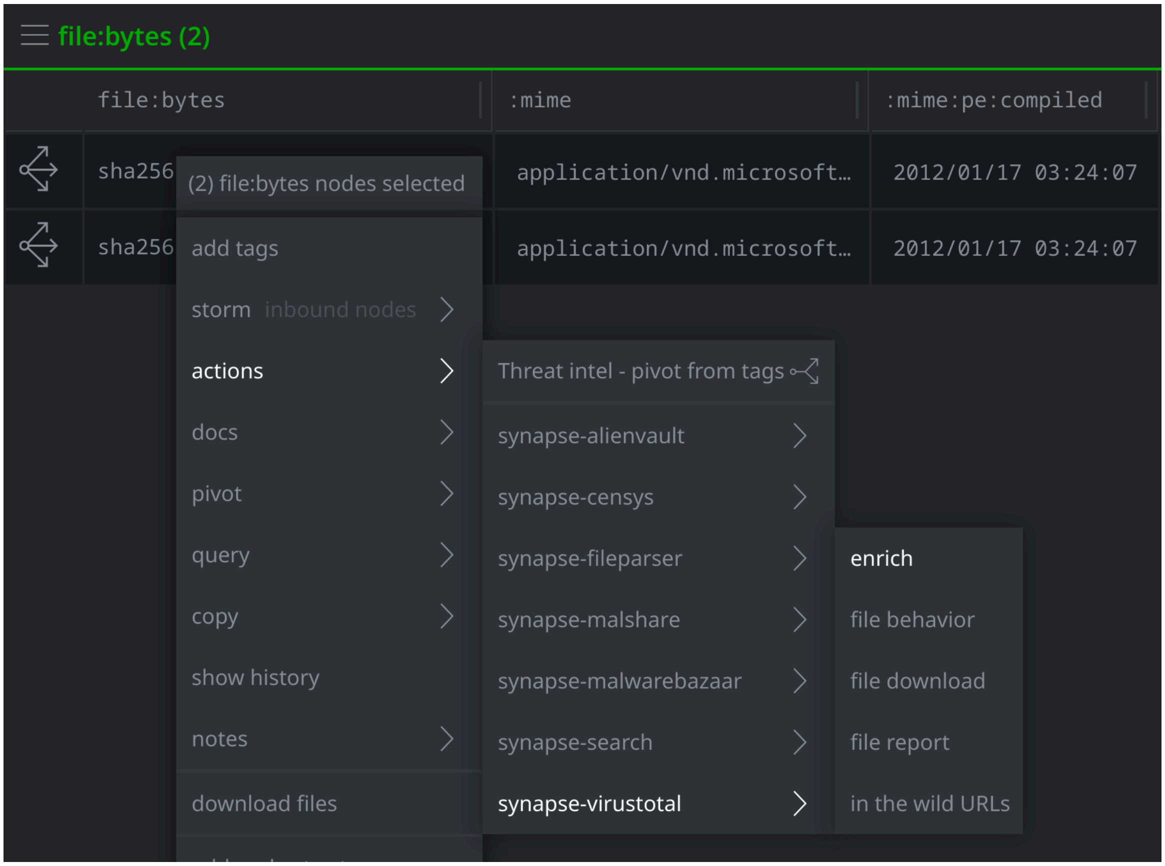
Part 2

We want some additional information about the files. We want to download any file reports that are available from VirusTotal.

- In your **Results Panel,** click the **hamburger menu** next to the **file:bytes** header and choose **Select all:**



- **Right-click** the files and select **actions > synapse-virustotal > enrich** to download the associated data from VirusTotal:

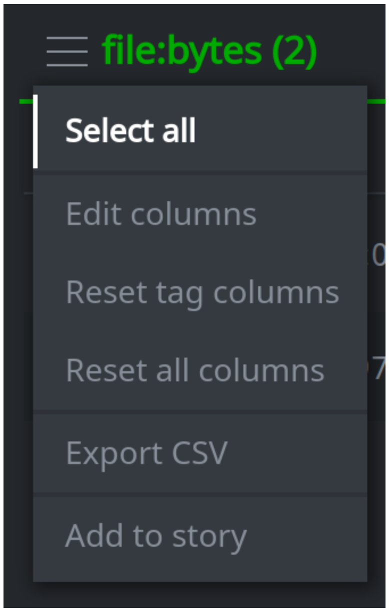


You may notice that we're using the "enrich" Node Action rather than the one named "file report". Although the two are similar, the "enrich" Node Action will return just the file report, while the "file report" Node Action will also return generic network execution data. We are using the enrich Node Action so that we can focus on just the file report.

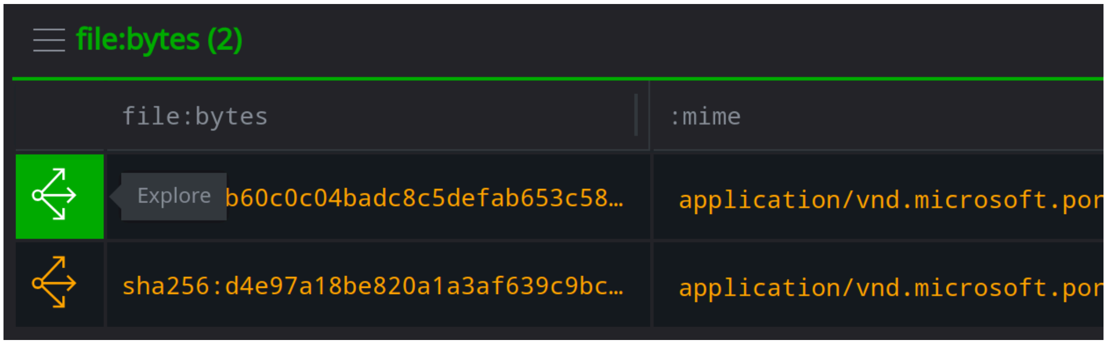
Question 5: Several tags were applied to the files when the VirusTotal reports were ingested. What do these tags tell us about the possible behavior or nature of the files?

Now we want to look at any malware detection (antivirus hits or YARA rules) that were returned with the VirusTotal reports.

- In your **Results Panel,** click the **hamburger menu** next to the **file:bytes** header and choose **Select all:**



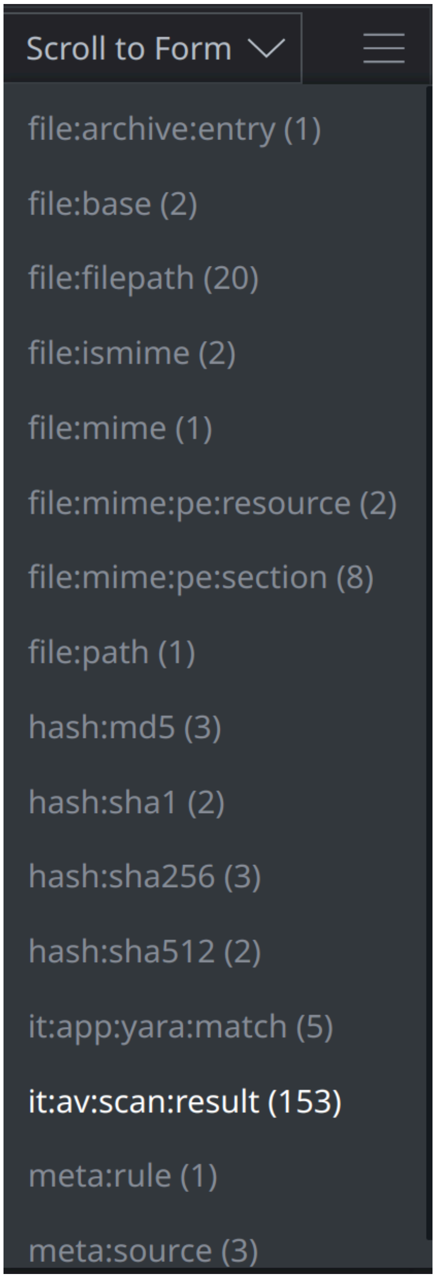
- Click the **Explore** button next to either file to display adjacent nodes:



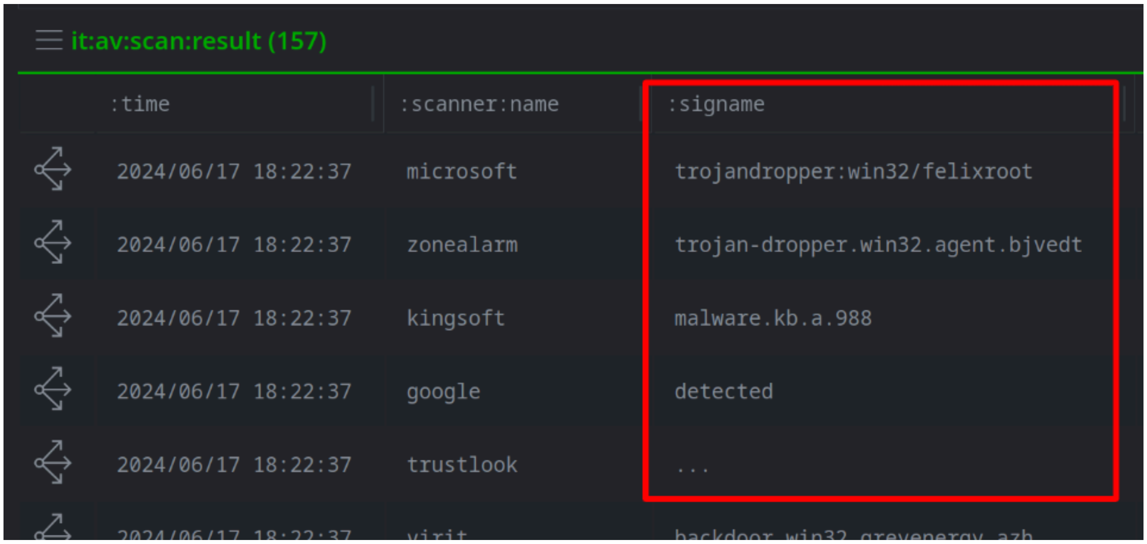
file:bytes (2)

	file:bytes	:mime
	b60c0c04badc8c5defab653c58…	application/vnd.microsoft.por
	sha256:d4e97a18be820a1a3af639c9bc…	application/vnd.microsoft.por

- To view the antivirus signature names, use **Scroll to Form** to navigate to the `it:av:scan:result` nodes:



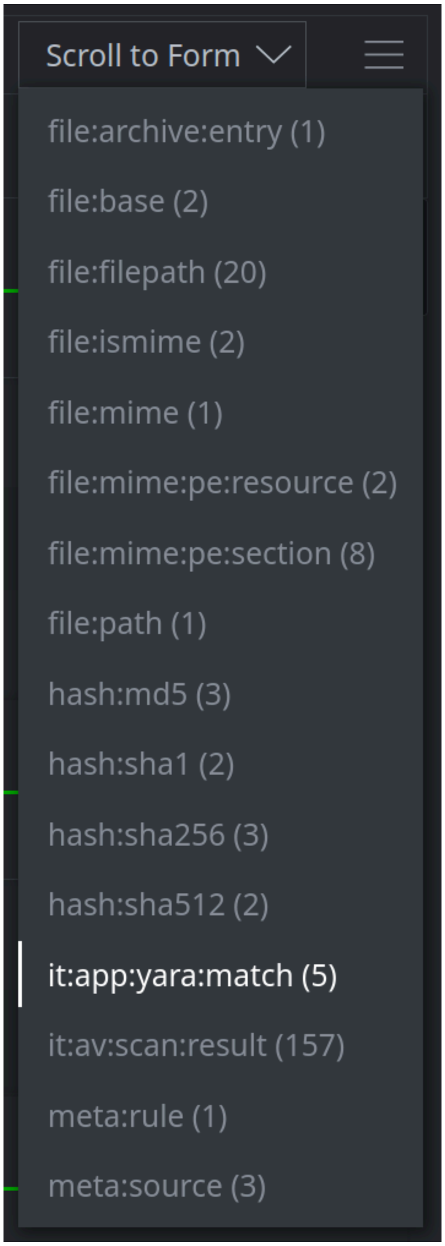
- Browse the signature names in the **:signame** column:



it:av:scan:result (157)

:time	:scanner:name	:signame
2024/06/17 18:22:37	microsoft	trojandropper:win32/felixroot
2024/06/17 18:22:37	zonealarm	trojan-dropper.win32.agent.bjvedt
2024/06/17 18:22:37	kingsoft	malware.kb.a.988
2024/06/17 18:22:37	google	detected
2024/06/17 18:22:37	trustlook	...
2024/06/17 18:22:37	virit	backdoor.win32.greyenergy.azh

- To view YARA rule names, use **Scroll to Form** to navigate to the `it:app:yara:match` nodes:



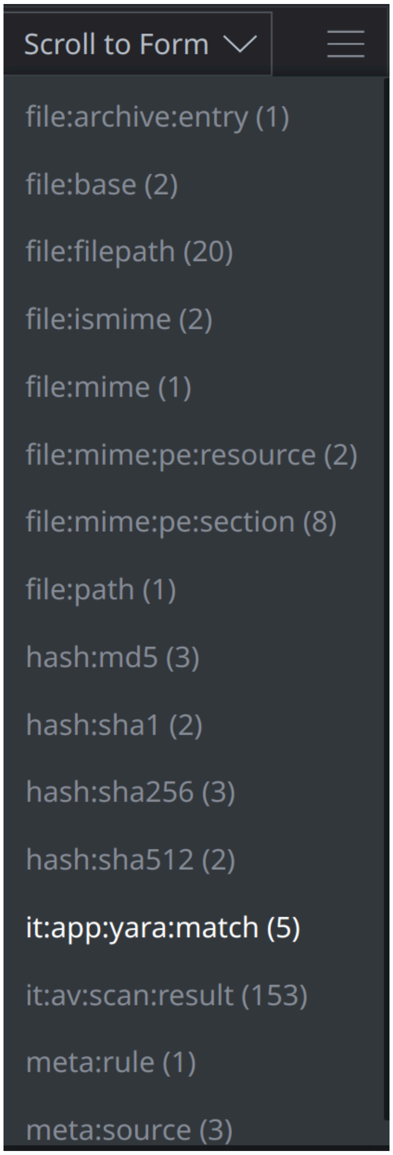
Scroll to Form

file:archive:entry (1)
file:base (2)
file:filepath (20)
file:ismime (2)
file:mime (1)
file:mime:pe:resource (2)
file:mime:pe:section (8)
file:path (1)
hash:md5 (3)
hash:sha1 (2)
hash:sha256 (3)
hash:sha512 (2)
it:app:yara:match (5)
it:av:scan:result (157)
meta:rule (1)
meta:source (3)

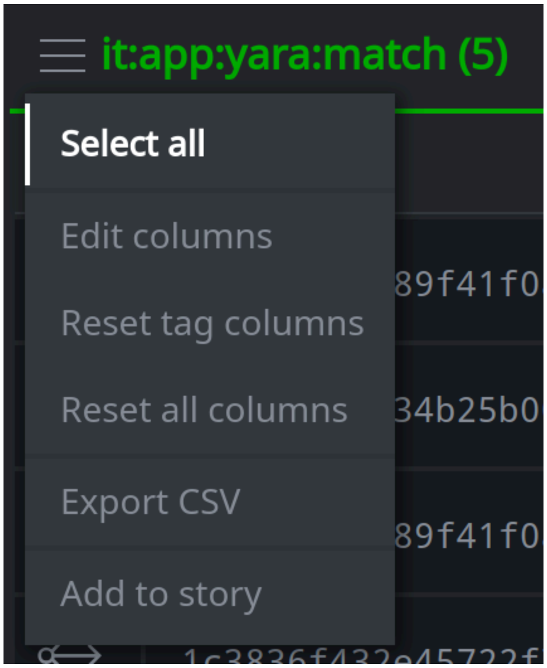
Question 6: Do any signature names or YARA rule names hint at a malware family for the file?

Some of the AV signatures and all of the YARA rules refer to "GreyEnergy". VirusTotal does not provide details on the AV signatures, so we can not tell how accurate they are. However, we may be able to look at the content of the YARA rules.

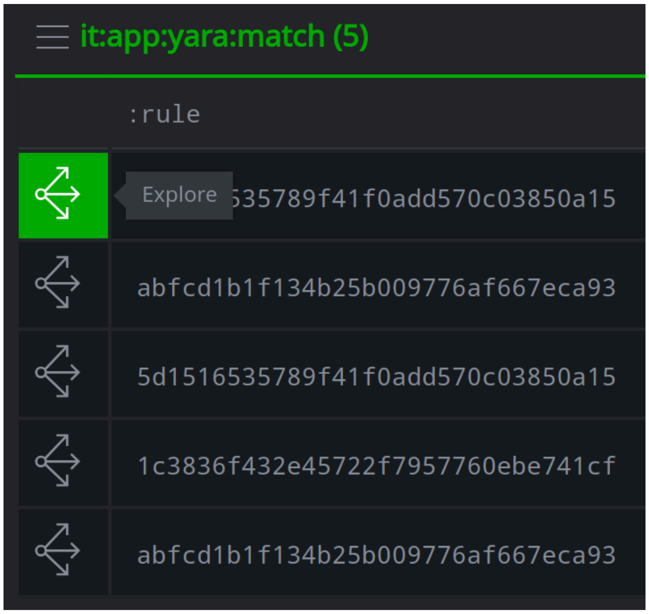
- Use **Scroll to Form** to navigate to the `it:app:yara:match` nodes:



- Click the **hamburger menu** next to the **it:app:yara:match** header and choose **Select All:**

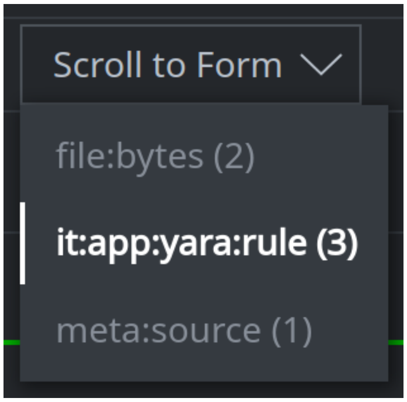


- Click the **Explore** button next to any selected node to navigate to adjacent nodes:

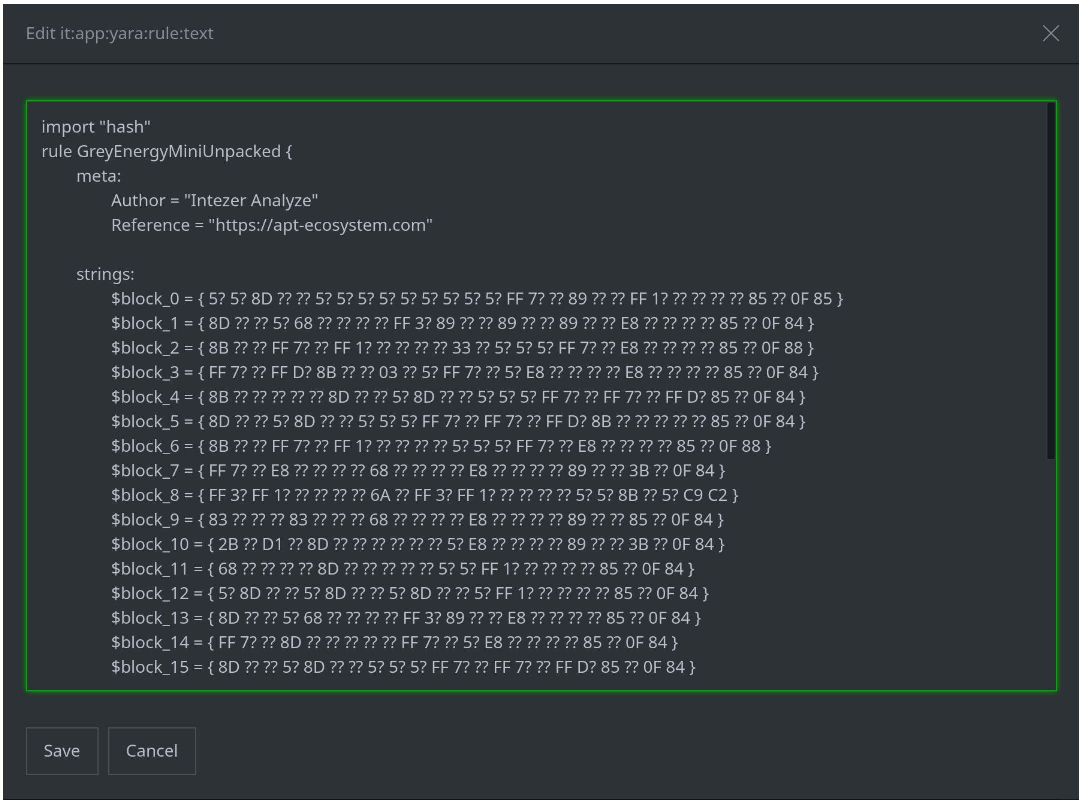


:rule
535789f41f0add570c03850a15
abfcd1b1f134b25b009776af667eca93
5d1516535789f41f0add570c03850a15
1c3836f432e45722f7957760ebe741cf
abfcd1b1f134b25b009776af667eca93

- Locate the `it:app:yara:rule` nodes (use **Scroll to Form** if necessary):



- For each rule, **hover over** the `:text` property to view the text of the rule (or **double-click** the `:text` property to open it in a larger window):



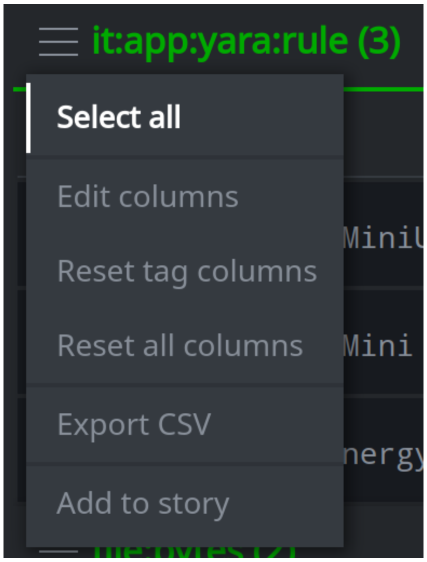
```
import "hash"
rule GreyEnergyMiniUnpacked {
    meta:
        Author = "Intezer Analyze"
        Reference = "https://apt-ecosystem.com"

    strings:
        $block_0 = { 5? 5? 8D ?? ?? 5? 5? 5? 5? 5? 5? 5? 5? 5? FF 7? ?? 89 ?? ?? FF 1? ?? ?? ?? ?? 85 ?? 0F 85 }
        $block_1 = { 8D ?? ?? 5? 68 ?? ?? ?? ?? FF 3? 89 ?? ?? 89 ?? ?? 89 ?? ?? E8 ?? ?? ?? ?? 85 ?? 0F 84 }
        $block_2 = { 8B ?? ?? FF 7? ?? FF 1? ?? ?? ?? ?? 33 ?? 5? 5? 5? FF 7? ?? E8 ?? ?? ?? ?? 85 ?? 0F 88 }
        $block_3 = { FF 7? ?? FF D? 8B ?? ?? 03 ?? 5? FF 7? ?? 5? E8 ?? ?? ?? ?? E8 ?? ?? ?? ?? 85 ?? 0F 84 }
        $block_4 = { 8B ?? ?? ?? ?? ?? 8D ?? ?? 5? 8D ?? ?? 5? 5? 5? FF 7? ?? FF 7? ?? FF D? 85 ?? 0F 84 }
        $block_5 = { 8D ?? ?? 5? 8D ?? ?? 5? 5? 5? FF 7? ?? FF 7? ?? FF D? 8B ?? ?? ?? ?? ?? 85 ?? 0F 84 }
        $block_6 = { 8B ?? ?? FF 7? ?? FF 1? ?? ?? ?? ?? 5? 5? 5? FF 7? ?? E8 ?? ?? ?? ?? 85 ?? 0F 88 }
        $block_7 = { FF 7? ?? E8 ?? ?? ?? ?? 68 ?? ?? ?? ?? E8 ?? ?? ?? ?? 89 ?? ?? 3B ?? 0F 84 }
        $block_8 = { FF 3? FF 1? ?? ?? ?? ?? 6A ?? FF 3? FF 1? ?? ?? ?? ?? 5? 5? 8B ?? 5? C9 C2 }
        $block_9 = { 83 ?? ?? ?? 83 ?? ?? ?? 68 ?? ?? ?? ?? E8 ?? ?? ?? ?? 89 ?? ?? 85 ?? 0F 84 }
        $block_10 = { 2B ?? D1 ?? 8D ?? ?? ?? ?? ?? ?? 5? E8 ?? ?? ?? ?? 89 ?? ?? 3B ?? 0F 84 }
        $block_11 = { 68 ?? ?? ?? ?? 8D ?? ?? ?? ?? ?? 5? 5? FF 1? ?? ?? ?? ?? 85 ?? 0F 84 }
        $block_12 = { 5? 8D ?? ?? 5? 8D ?? ?? 5? 8D ?? ?? 5? FF 1? ?? ?? ?? ?? 85 ?? 0F 84 }
        $block_13 = { 8D ?? ?? 5? 68 ?? ?? ?? ?? FF 3? 89 ?? ?? E8 ?? ?? ?? ?? 85 ?? 0F 84 }
        $block_14 = { FF 7? ?? 8D ?? ?? ?? ?? ?? FF 7? ?? 5? E8 ?? ?? ?? ?? 85 ?? 0F 84 }
        $block_15 = { 8D ?? ?? 5? 8D ?? ?? 5? 5? 5? FF 7? ?? FF 7? ?? FF D? 85 ?? 0F 84 }
```

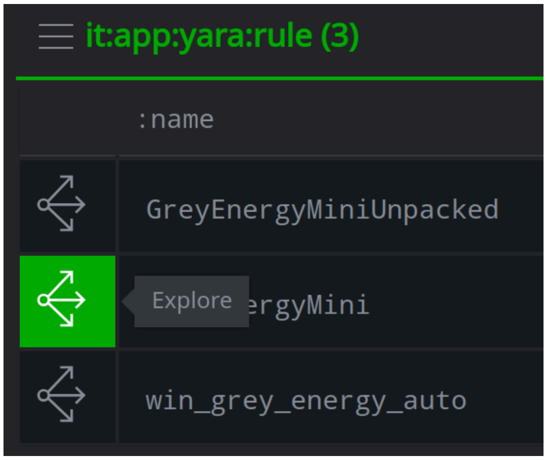
Question 7: Who is the author or authors of the rules? Do the rules seem very broad or are they very specific?

Some YARA rules used by third-party vendors (such as VirusTotal) may come from public reports or repositories. You want to see if these YARA rules have an associated source.

- In your **Results Panel,** click the **hamburger menu** next to the **it:app:yara:rule** header and choose **Select All:**

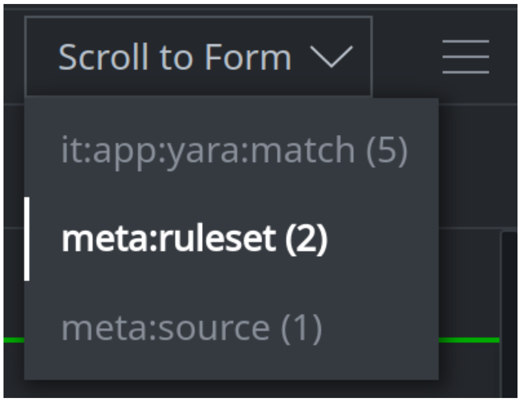


- Click the **Explore** button next to any selected node to navigate to adjacent nodes:



:name
GreyEnergyMiniUnpacked
ergyMini
win_grey_energy_auto

- Locate the `meta:ruleset` nodes (use **Scroll to Form** if necessary):



Question 8: Are the YARA rules associated with any rulesets? If so, where can you find the rulesets?

Part 3

Based on what we have seen, there is a good chance that these files belong to the "GreyEnergy" malware family.

We could look for other files that match these **YARA rules.** Because the Synapse YARA Grid Power-Up is not installed in our Bootcamp demo instances, we will look at antivirus signatures instead.

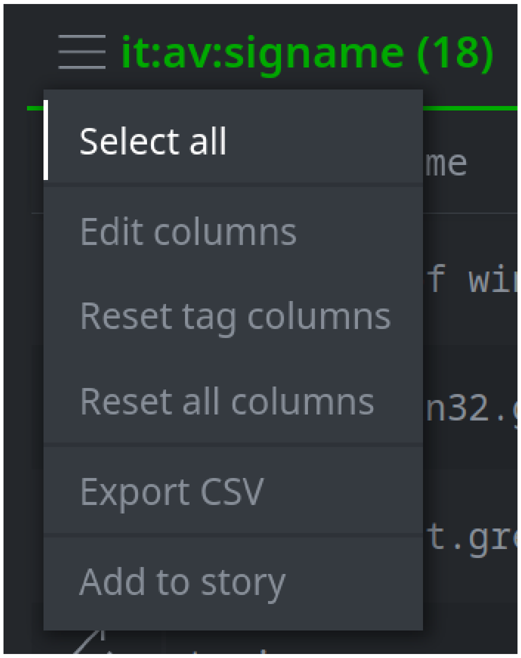
Many antivirus signature names are very specific (such as **backdoor.win32.greyenergy.azh**). We want to find files that match **any** signature that **contains** the string "greyenergy".

- Enter the following in your **Storm Query Bar** and press **Enter** to run the query:

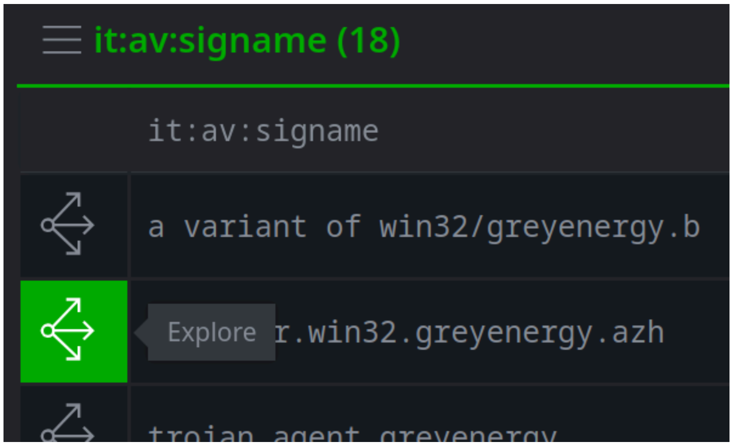
```
it:av:signame~=greyenergy
```

Question 9: How many signatures did you find?

- In the **Results Panel,** click the **hamburger menu** to the left of the **it:av:signame** table header and choose **Select All** to select all of the `it:av:signame` nodes:

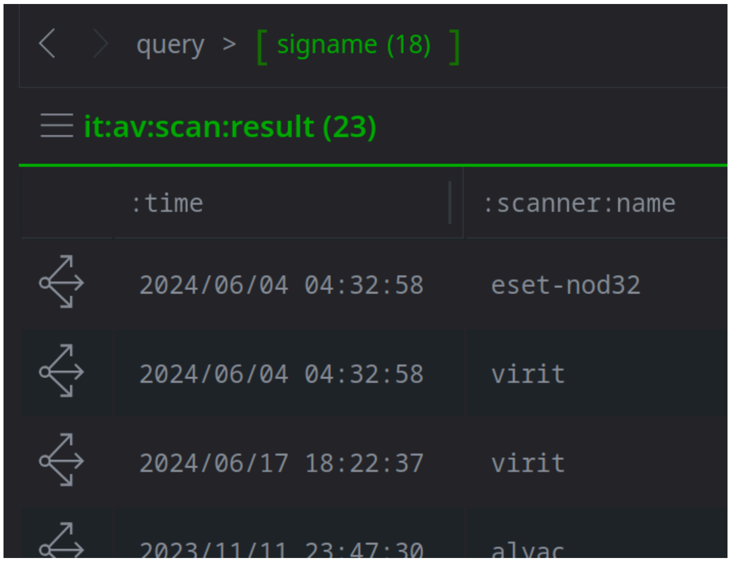


- Click the **Explore** button next to any of the nodes to display adjacent nodes:


it:av:signame (18)

it:av:signame
a variant of win32/greyenergy.b
r.win32.greyenergy.azh
trojan agent greyenergy

- Locate the **`it:av:scan:result`** nodes:

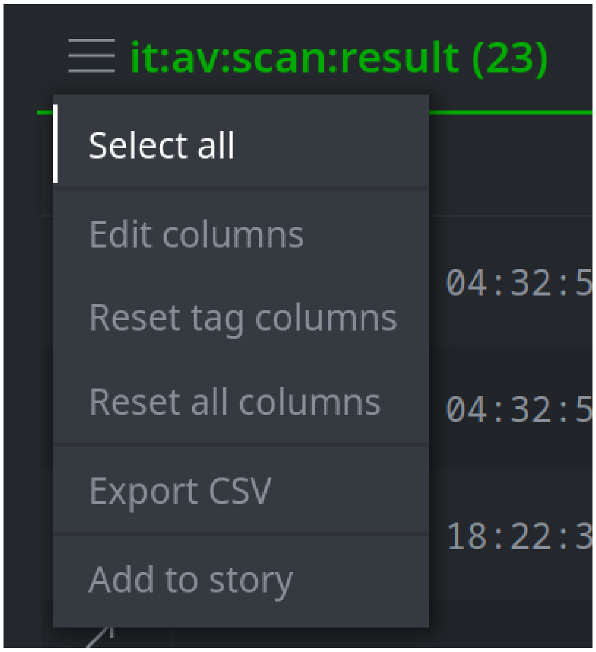


query > [signame (18)]

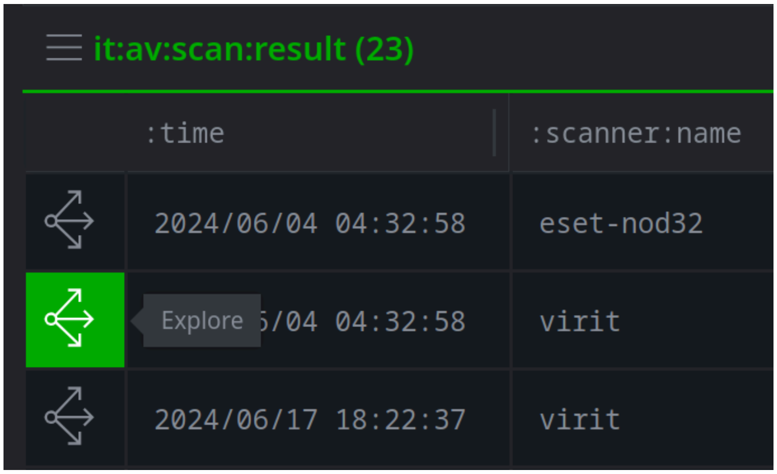
it:av:scan:result (23)

:time	:scanner:name
2024/06/04 04:32:58	eset-nod32
2024/06/04 04:32:58	virit
2024/06/17 18:22:37	virit
2023/11/11 23:47:30	alvac

- Click the **hamburger menu** to the left of the **it:av:scan:result** table header and choose **Select All** to select all of the **`it:av:scan:result`** nodes:



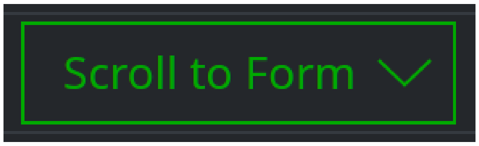
- Click the **Explore** button next to any of the nodes to display adjacent nodes:



it:av:scan:result (23)

	:time	:scanner:name
	2024/06/04 04:32:58	eset-nod32
	Explore 5/04 04:32:58	virit
	2024/06/17 18:22:37	virit

- Locate the **`file:bytes`** nodes in your results (use **Scroll to Form** if necessary):



Question 10: How many files are detected by one or more of the greyenergy signatures?

Exercise 2

Objective:

- **Use code signing certificate data extracted by the FileParser Power-Up to search for other files signed with the same certificate.**

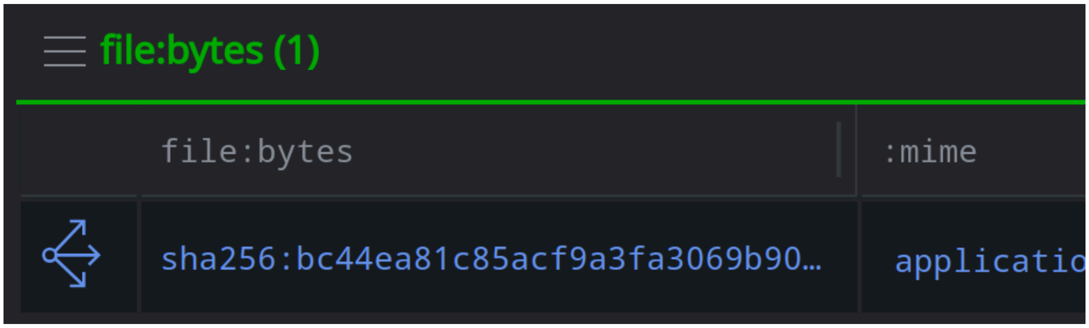
You are investigating a malicious file. You have downloaded the file, parsed it with the FileParser Power-Up, and retrieved the associated file report from VirusTotal.

- Enter the following into the **Storm Query Bar** and press **Enter** to run the query and view the file:

```
file:bytes=bc44ea81c85acf9a3fa3069b90aa4c2286f2813da2240cafa8b2
ad6ac997fe56
```

> **Note:** The exercise PDFs may insert line breaks or spaces where values (such as the SHA256, above) are forced to wrap. If you copy the above into your Storm query bar and the query fails to run, you may need to manually remove the space / line break.

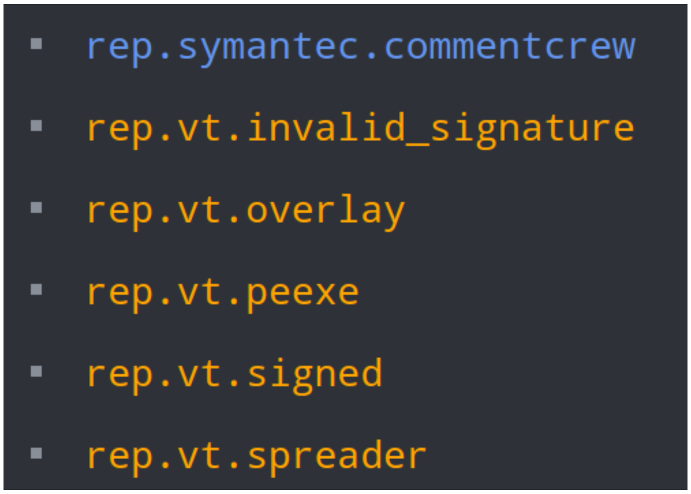
- In the **Results Panel, select** the file:



file:bytes (1)

file:bytes	:mime
sha256:bc44ea81c85acf9a3fa3069b90…	applicatio

- In the **Details Panel,** view information about the file. VirusTotal applied tags to this file related to a digital signature:

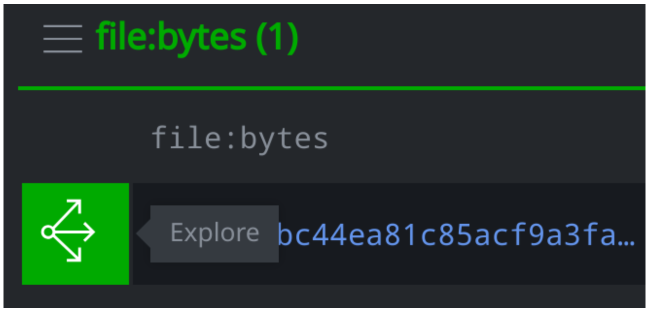


- rep.symantec.commentcrew
- rep.vt.invalid_signature
- rep.vt.overlay
- rep.vt.peexe
- rep.vt.signed
- rep.vt.spreader

Question 1: What do the tags imply about this file?

> You know that FileParser can extract and model code signing certificates. You want to view the certificate details that FileParser identified.

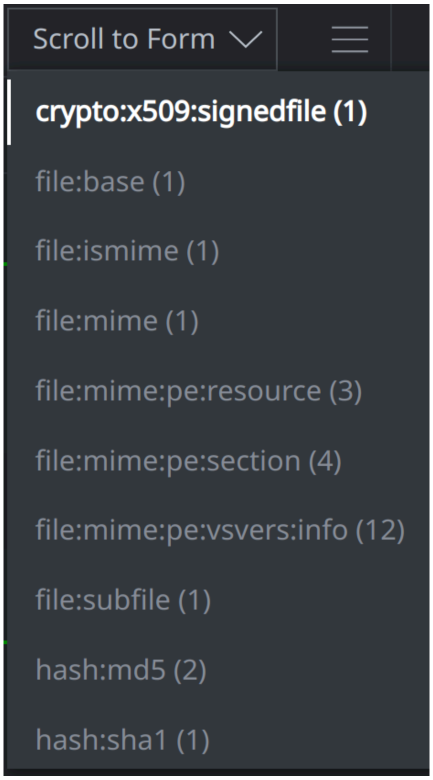
- Click the **Explore** button next to the `file:bytes` node to display adjacent nodes:


file:bytes (1)

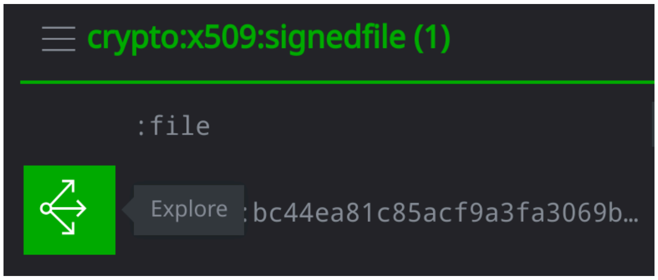
file:bytes

bc44ea81c85acf9a3fa…

- Click the **Scroll to Form** button to navigate to the `crypto:x509:signedfile` node:



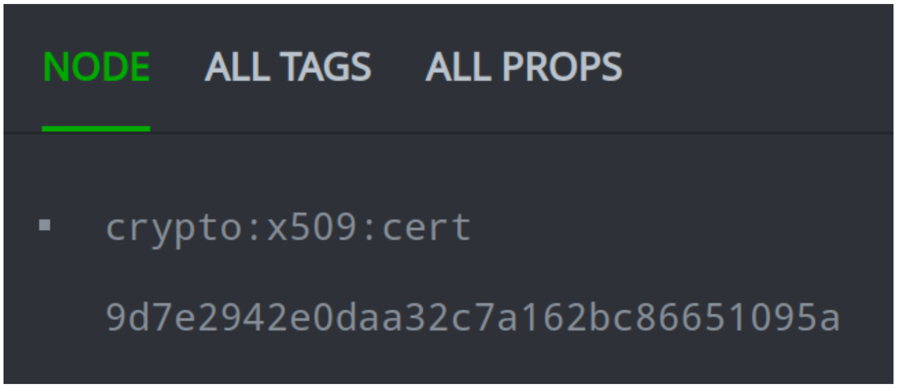
- Click the **Explore** button next to the `crypto:x509:signedfile` node to display adjacent nodes:


crypto:x509:signedfile (1)

:file

:bc44ea81c85acf9a3fa3069b…

- Locate and **select** the **`crypto:x509:cert`** node. View the node's properties in the **Details Panel:**



NODE ALL TAGS ALL PROPS

▪ crypto:x509:cert

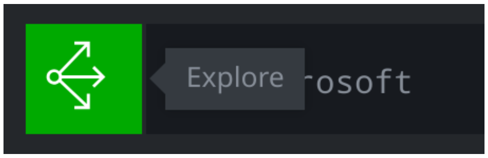
9d7e2942e0daa32c7a162bc86651095a

Question 2: What are the Subject and Issuer of the certificate?

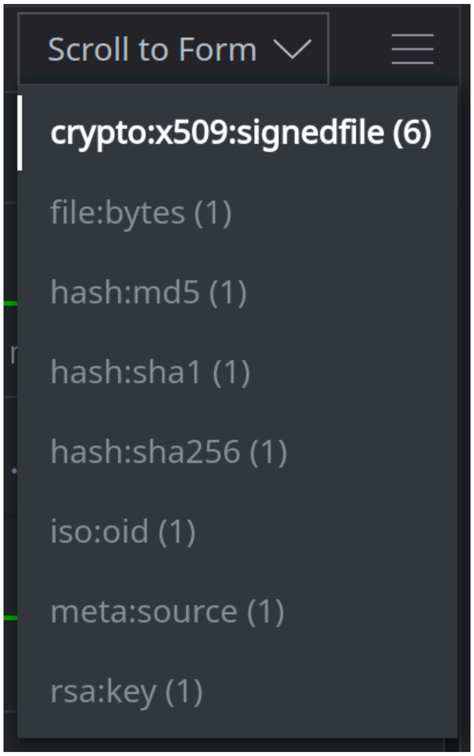
Question 3: What is the validity period for the certificate?

The certificate looks suspicious. You want to know if there are other files signed with the same certificate in Synapse.

- Click the **Explore** button next to the **`crypto:x509:cert`** node:



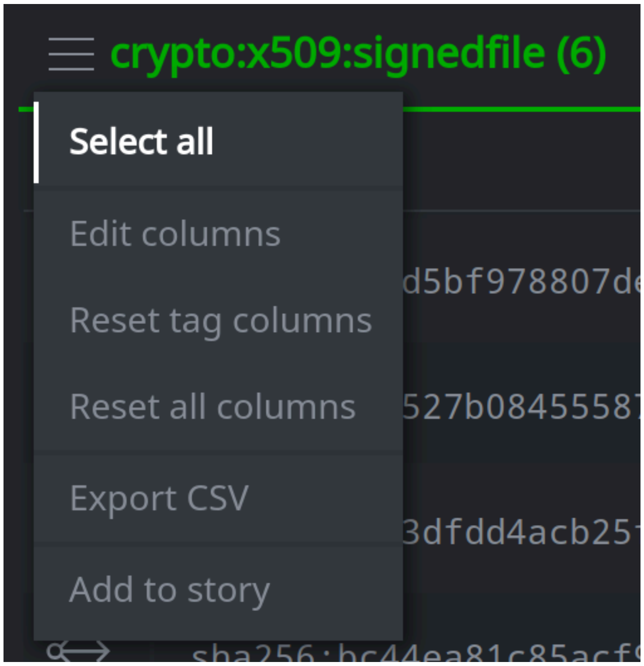
- Locate the **`crypto:x509:signedfile`** nodes in your results (use **Scroll to Form** if necessary):



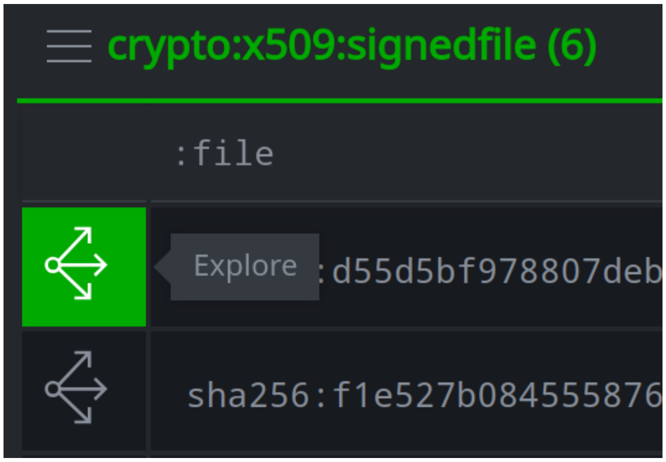
Question 4: How many files were signed with this certificate?

You want to see what (if anything) we know about the files that were signed with this certificate.

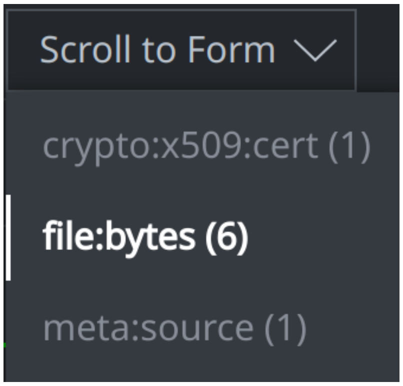
- In the **Results Panel,** click the **hamburger menu** to the left of the **crypto:x509:signedfile** table header and choose **Select All:**



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



- Locate the **`file:bytes`** nodes signed with the certificate (use **Scroll to Form** if necessary):



Question 5: How many files have tags that show they are associated with a malware family or threat group?

Question 6: Did you identify any "unknown" (untagged) files signed with the same certificate?