Synapse Bootcamp - Module 15

Static Malware Analysis - Answer Key

Answer Key

Static Malware Analysis

Exercise 1 Answer

Objective:

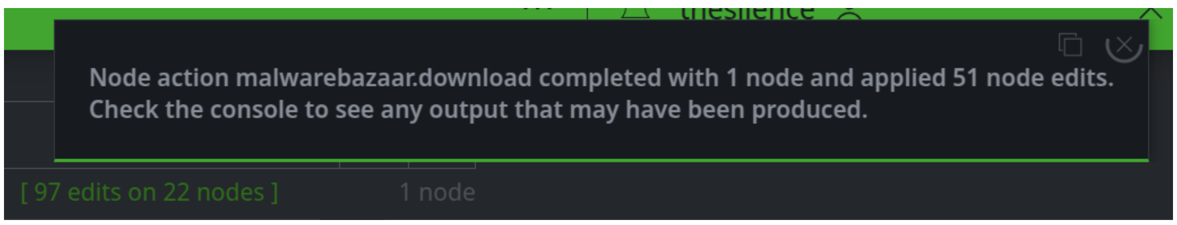
- **Use Power-Ups to enrich and research a suspicious file.**
- **Examine static data to gain insight into the file.**

Part 1

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

- You should have been able to download the file from MalwareBazaar.

  Synapse displays a status message in the upper right of your browser as well as a summary of the changes (edits) made when the file was downloaded and parsed (e.g., **[97 edits on 22 nodes]**):

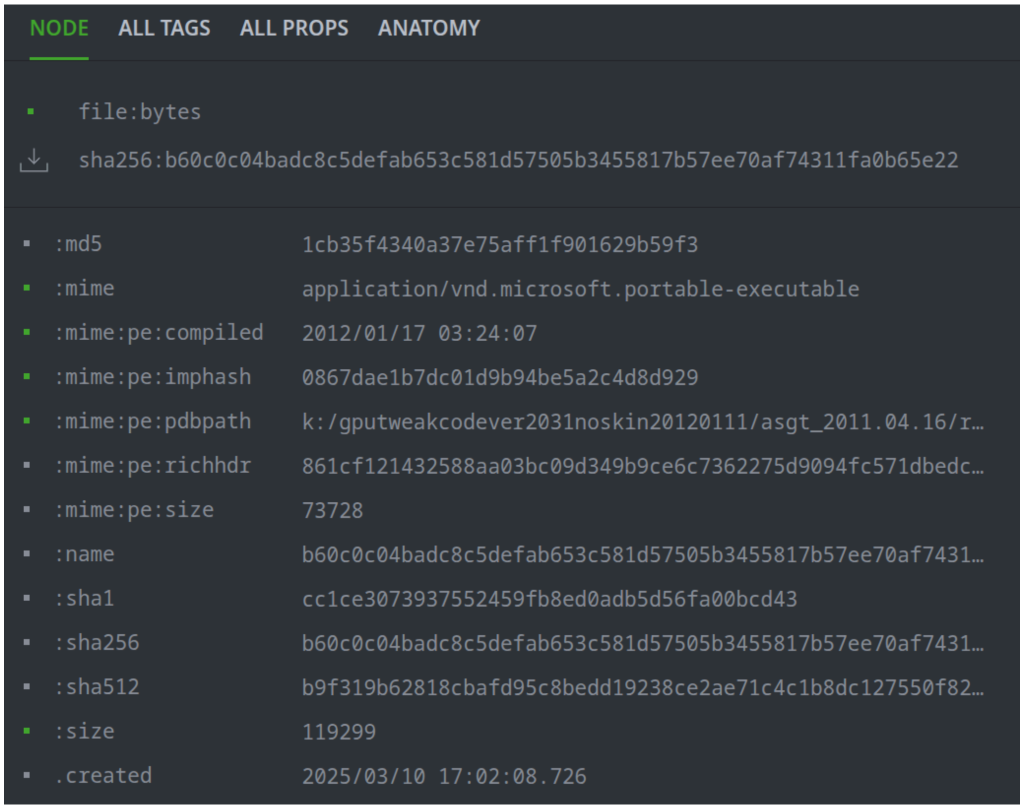


```
Node action malwarebazaar.download completed with 1 node and applied 51 node edits.
Check the console to see any output that may have been produced.
```

```
[ 97 edits on 22 nodes ]          1 node
```

Question 2: What properties are set for the file?

- The file's size, additional hash values (MD5, SHA1), mime type, and PE-specific metadata are set:



```
file:bytes
sha256:b60c0c04badc8c5defab653c581d57505b3455817b57ee70af74311fa0b65e22

:md5                1cb35f4340a37e75aff1f901629b59f3
:mime               application/vnd.microsoft.portable-executable
:mime:pe:compiled   2012/01/17 03:24:07
:mime:pe:imphash    0867dae1b7dc01d9b94be5a2c4d8d929
:mime:pe:pdbpath    k:/gputweakcodever2031noskin20120111/asgt_2011.04.16/r…
:mime:pe:richhdr    861cf121432588aa03bc09d349b9ce6c7362275d9094fc571dbedc…
:mime:pe:size       73728
:name               b60c0c04badc8c5defab653c581d57505b3455817b57ee70af7431…
:sha1               cc1ce3073937552459fb8ed0adb5d56fa00bcd43
:sha256             b60c0c04badc8c5defab653c581d57505b3455817b57ee70af7431…
:sha512             b9f319b62818cbafd95c8bedd19238ce2ae71c4c1b8dc127550f82…
:size               119299
.created            2025/03/10 17:02:08.726
```

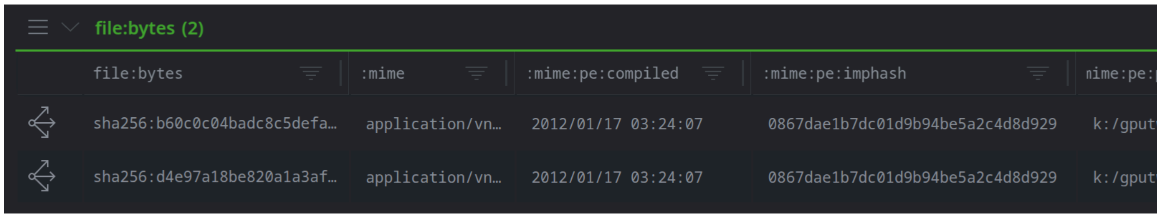
Synapse calculates and sets a file's size and hash values when the file is downloaded (or uploaded) into Synapse's Axon storage.

In addition, Synapse-MalwareBazaar (and some other Power-Ups that download malware samples) use **Synapse-FileParser** to parse the file and extract additional information such as the compile time, PDB path, PE import hash, etc.

(**Note** that as of August 2025, the Synapse-VirusTotal Power-Up does **not** automatically parse downloaded samples with Synapse-FileParser; if you want to parse files retrieved from VirusTotal, you need to do it manually, or configure automation to do it for you.)

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

- There is **one additional** file (two files in total) with the same import hash:



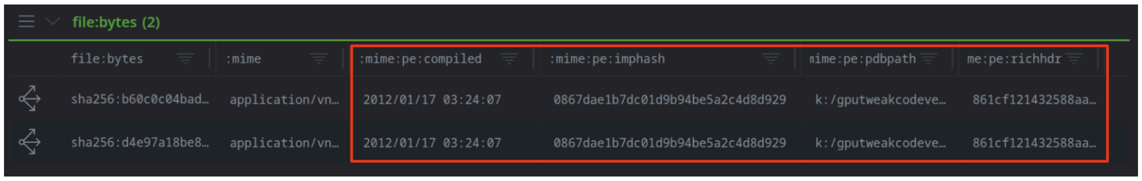
file:bytes (2)

	file:bytes	:mime	:mime:pe:compiled	:mime:pe:imphash	nime:pe:
	sha256:b60c0c04badc8c5defa…	application/vn…	2012/01/17 03:24:07	0867dae1b7dc01d9b94be5a2c4d8d929	k:/gput
	sha256:d4e97a18be820a1a3af…	application/vn…	2012/01/17 03:24:07	0867dae1b7dc01d9b94be5a2c4d8d929	k:/gput

---

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

- The files also share the same compile time, PDB path, and Rich Header hash:



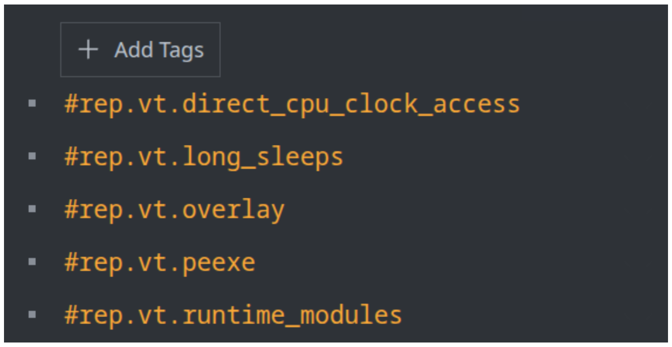
file:bytes (2)

	file:bytes	:mime	:mime:pe:compiled	:mime:pe:imphash	nime:pe:pdbpath	me:pe:richhdr
	sha256:b60c0c04bad…	application/vn…	2012/01/17 03:24:07	0867dae1b7dc01d9b94be5a2c4d8d929	k:/gputweakcodeve…	861cf121432588aa…
	sha256:d4e97a18be8…	application/vn…	2012/01/17 03:24:07	0867dae1b7dc01d9b94be5a2c4d8d929	k:/gputweakcodeve…	861cf121432588aa…

Part 2

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?

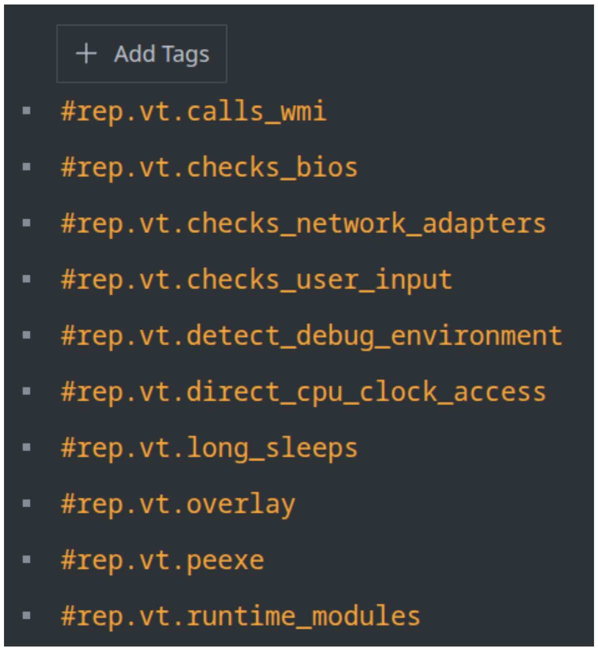
- The VirusTotal tags indicate that the files:
  - Are PE executables (**`rep.vt.peexe`**)
  - Contain appended data at the end of the file (**`rep.vt.overlay`**)
  - Have a long pause or 'sleep' during execution (**`rep.vt.long_sleeps`**)
  - Access the CPU clock (**`rep.vt.direct_cpu_clock_access`**)
  - May load modules dynamically / at runtime (**`rep.vt.runtime_modules`**)



- #rep.vt.direct_cpu_clock_access
- #rep.vt.long_sleeps
- #rep.vt.overlay
- #rep.vt.peexe
- #rep.vt.runtime_modules

- One file has additional tags that show that it may:
  - Check network adapters (**`rep.vt.checks_network_adapters`**)

- Check for user activity (**`rep.vt.checks_user_input`**)
  - Check the system BIOS (**`rep.vt.checks_bios`**)
  - Checks if it is running in a sandbox / debug environment (**`rep.vt.detect_debug_environment`**)
  - Calls the Windows Management Instrumentation interface (**`rep.vt.calls_wmi`**)



- #rep.vt.calls_wmi
- #rep.vt.checks_bios
- #rep.vt.checks_network_adapters
- #rep.vt.checks_user_input
- #rep.vt.detect_debug_environment
- #rep.vt.direct_cpu_clock_access
- #rep.vt.long_sleeps
- #rep.vt.overlay
- #rep.vt.peexe
- #rep.vt.runtime_modules

The VirusTotal tags provide basic information about what a file "does" (or might do). They do not necessarily mean that the files are malicious.

However, some of the behaviors may be suspicious - things like sleeping during execution, checking whether any debugging programs are running, or checking for user input (such as mouse movement) can be used to detect whether the file is running in a sandbox environment. Some files may stop executing or change their behavior if a sandbox is detected in order to avoid analysis.

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

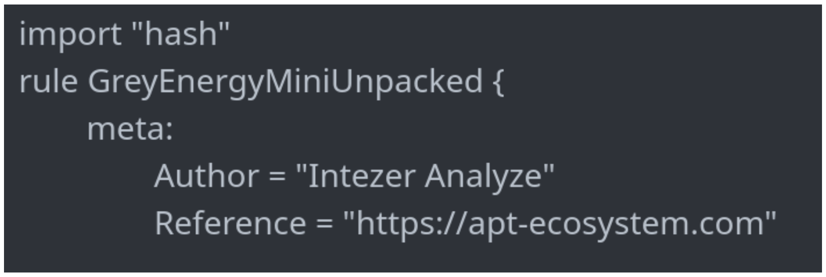
- Some signature names reference terms such as **felixroot, fragtor, greyenergy,** or **sandworm.** However, many signature names are generic and not very helpful.

- The YARA rule names all reference **greyenergy.**

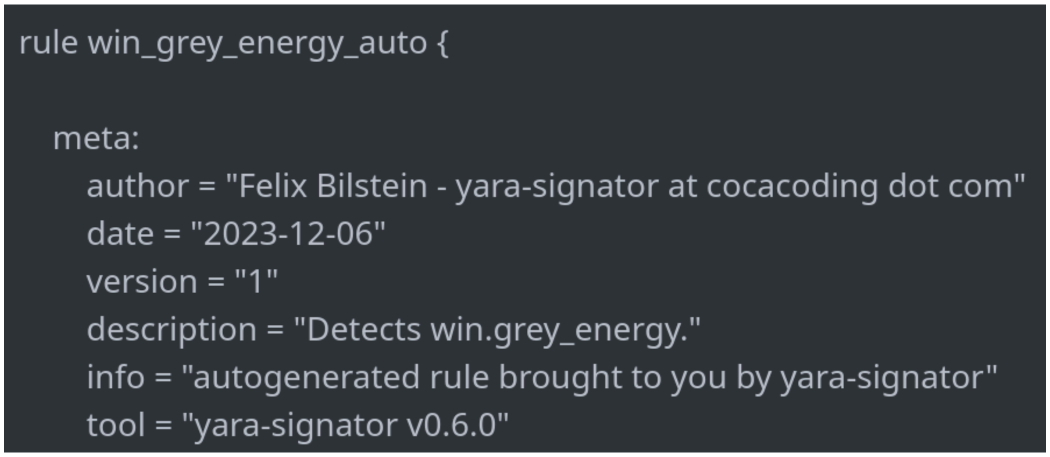
The signature names do not prove that the file belongs to one of these families, but the names may provide a starting point for additional research.

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

- Based on the YARA metadata, two rules were provided by **Intezer Analyze** and one was provided by **Felix Blistein** (this rule was automatically generated by the **YARA-Signator** tool):



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



```
rule win_grey_energy_auto {

    meta:
        author = "Felix Bilstein - yara-signator at cocacoding dot com"
        date = "2023-12-06"
        version = "1"
        description = "Detects win.grey_energy."
        info = "autogenerated rule brought to you by yara-signator"
        tool = "yara-signator v0.6.0"
```

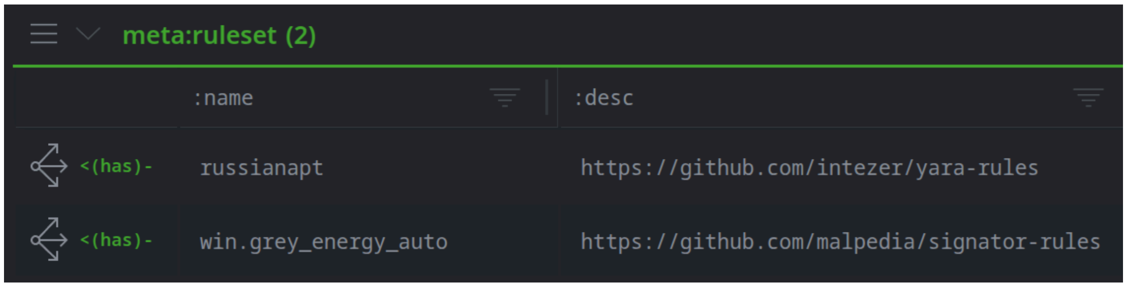
- The YARA rules are **specific**, looking for sequences of byte patterns in the PE executable files.

If you are familiar with YARA, the ability to view the **content** of a rule may help you decide how accurate or reliable you think the detection is. What does the rule look for? Does it search for some common strings or is it more specific?

If you do not know much about YARA, the **author** or **source** of a rule may affect how much you trust the rule. Is the rule provided by a company you know, or a security researcher you trust?

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

- The YARA rules are associated with **two** rulesets:
    - "russianapt" (from security firm Intezer)
    - "win.grey_energy_auto" (from Malpedia)


meta:ruleset (2)

	:name	:desc
<(has)-	russianapt	https://github.com/intezer/yara-rules
<(has)-	win.grey_energy_auto	https://github.com/malpedia/signator-rules

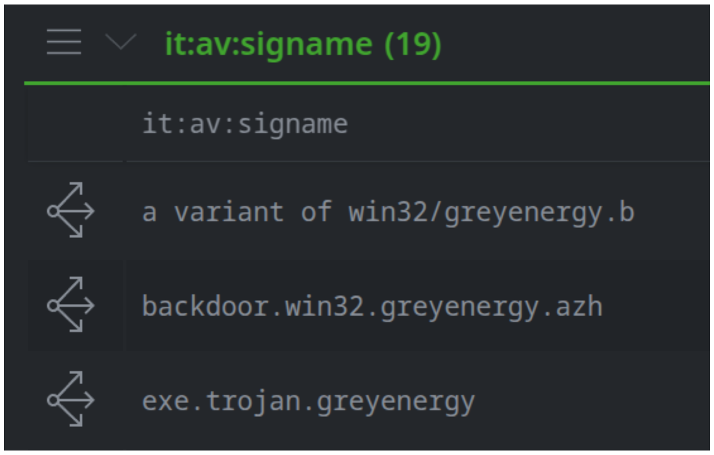
- Both rulesets are available from **Github:**
    - https://github.com/intezer/yara-rules
    - https://github.com/malpedia/signator-rules

If the "GreyEnergy" YARA rules seem useful and reliable, you can review additional rules from these authors / rulesets, and load the rules into Synapse.

Part 3

Question 9: How many signatures did you find?

- There are **19** AV signature (`it:av:signame`) nodes whose name contains the string **greyenergy** (as of August 2025):



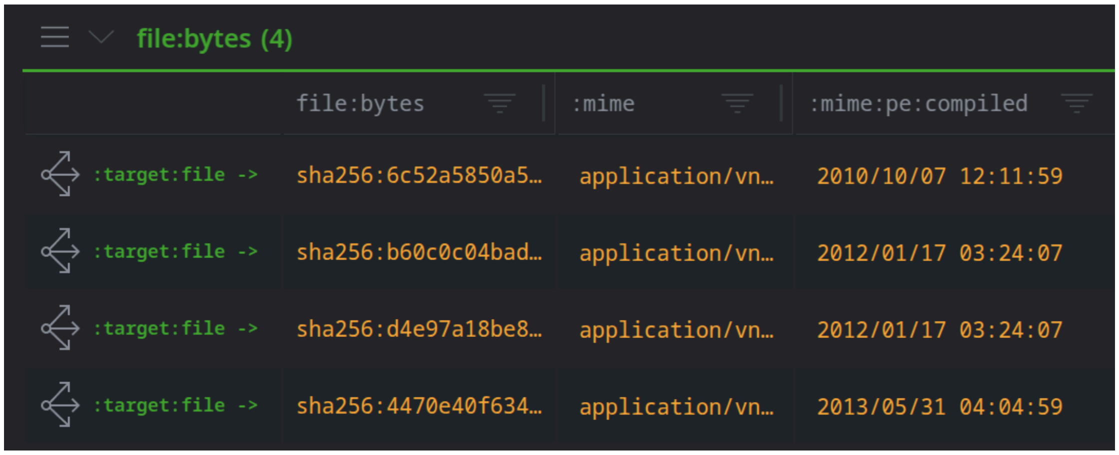
it:av:signame (19)

	it:av:signame
	a variant of win32/greyenergy.b
	backdoor.win32.greyenergy.azh
	exe.trojan.greyenergy

Note: Your answer may vary based on changes to external (e.g., VirusTotal) data and / or changes made to the data in your demo instance during this course.

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

- **Four** files are detected by the signatures (as of August 2025):



file:bytes (4)

	file:bytes	:mime	:mime:pe:compiled
:target:file ->	sha256:6c52a5850a5…	application/vn…	2010/10/07 12:11:59
:target:file ->	sha256:b60c0c04bad…	application/vn…	2012/01/17 03:24:07
:target:file ->	sha256:d4e97a18be8…	application/vn…	2012/01/17 03:24:07
:target:file ->	sha256:4470e40f634…	application/vn…	2013/05/31 04:04:59

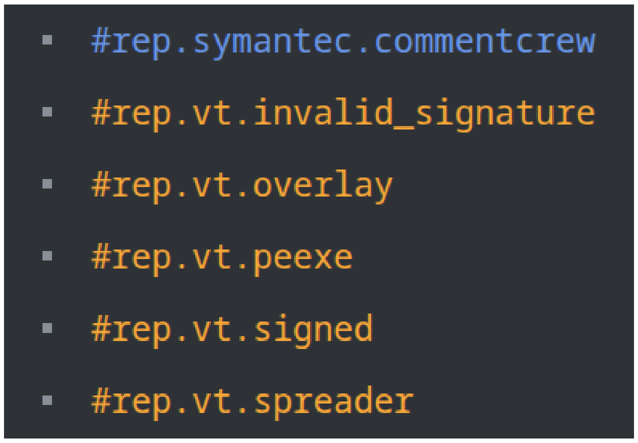
Note: Your answer may vary based on changes to external (e.g., VirusTotal) data and / or changes made to the data in your demo instance during this course.

Exercise 2 Answer

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

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

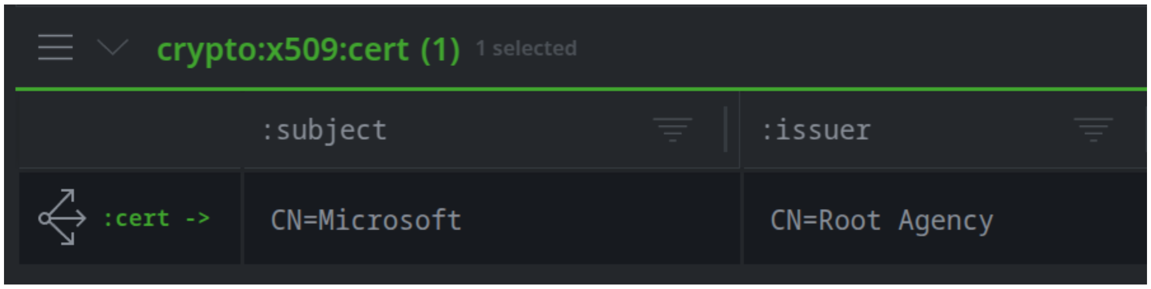
- The tags imply that:
  - the file was signed with a code-signing certificate (`rep.vt.signed`), but
  - the signature is invalid (`rep.vt.invalid_signature`).



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

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

- The Subject is **Microsoft** and the Issuer is **Root Agency:**

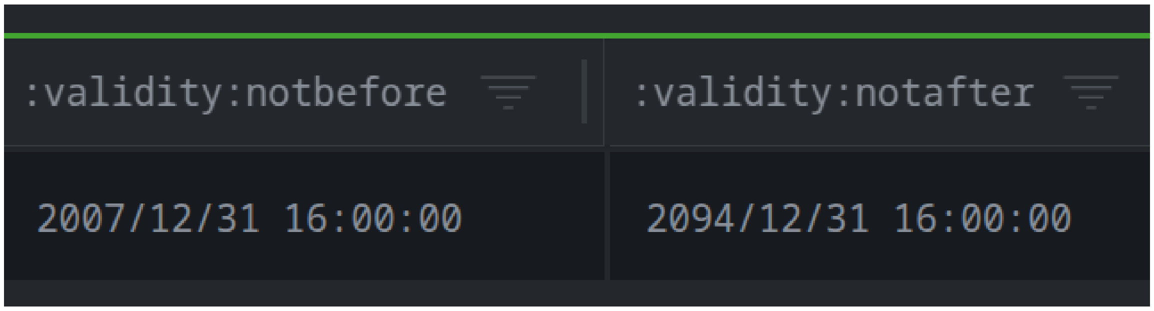


crypto:x509:cert (1) 1 selected

	:subject	:issuer
:cert ->	CN=Microsoft	CN=Root Agency

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

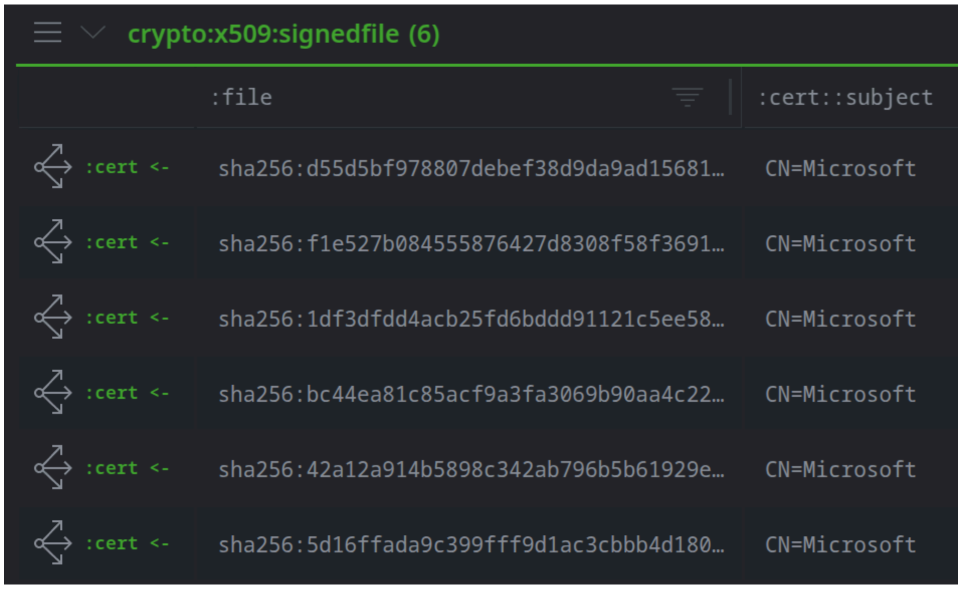
- The certificate is valid from **December 31, 2007** through **December 31, 2094:**



:validity:notbefore	:validity:notafter
2007/12/31 16:00:00	2094/12/31 16:00:00

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

- **Six** files were signed with this certificate:



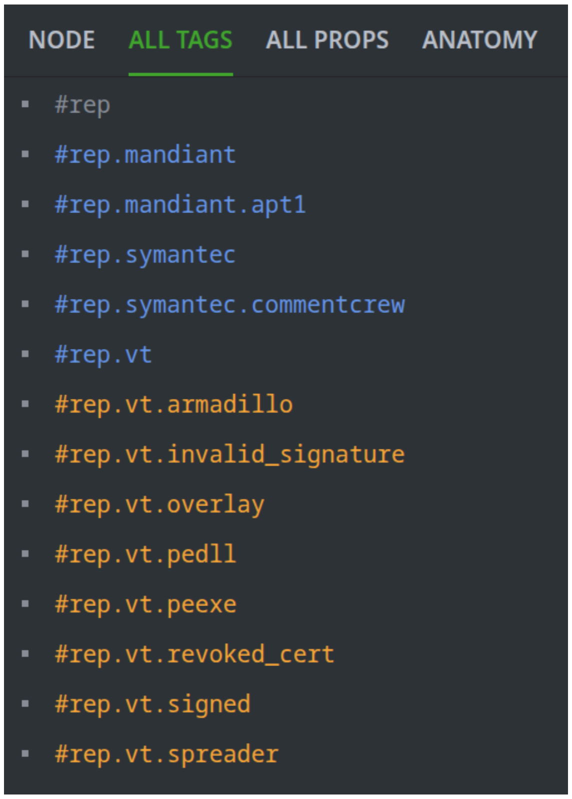
crypto:x509:signedfile (6)

	:file	:cert::subject
:cert <-	sha256:d55d5bf978807debef38d9da9ad15681…	CN=Microsoft
:cert <-	sha256:f1e527b084555876427d8308f58f3691…	CN=Microsoft
:cert <-	sha256:1df3dfdd4acb25fd6bddd91121c5ee58…	CN=Microsoft
:cert <-	sha256:bc44ea81c85acf9a3fa3069b90aa4c22…	CN=Microsoft
:cert <-	sha256:42a12a914b5898c342ab796b5b61929e…	CN=Microsoft
:cert <-	sha256:5d16ffada9c399fff9d1ac3cbbb4d180…	CN=Microsoft

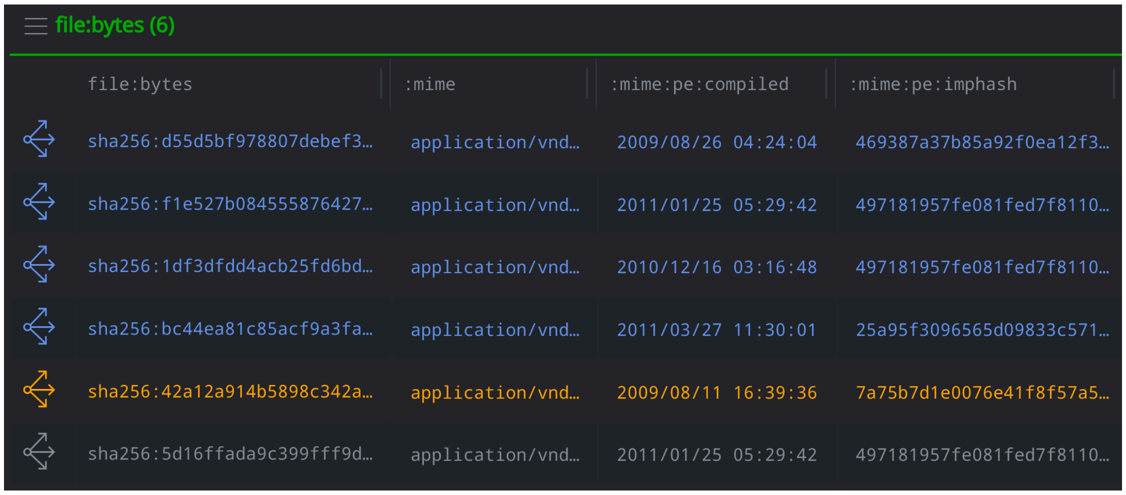
---

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

- **Four** files (in **blue**) have tags that associate them with a threat group:



- #rep
- #rep.mandiant
- #rep.mandiant.apt1
- #rep.symantec
- #rep.symantec.commentcrew
- #rep.vt
- #rep.vt.armadillo
- #rep.vt.invalid_signature
- #rep.vt.overlay
- #rep.vt.pedll
- #rep.vt.peexe
- #rep.vt.revoked_cert
- #rep.vt.signed
- #rep.vt.spreader

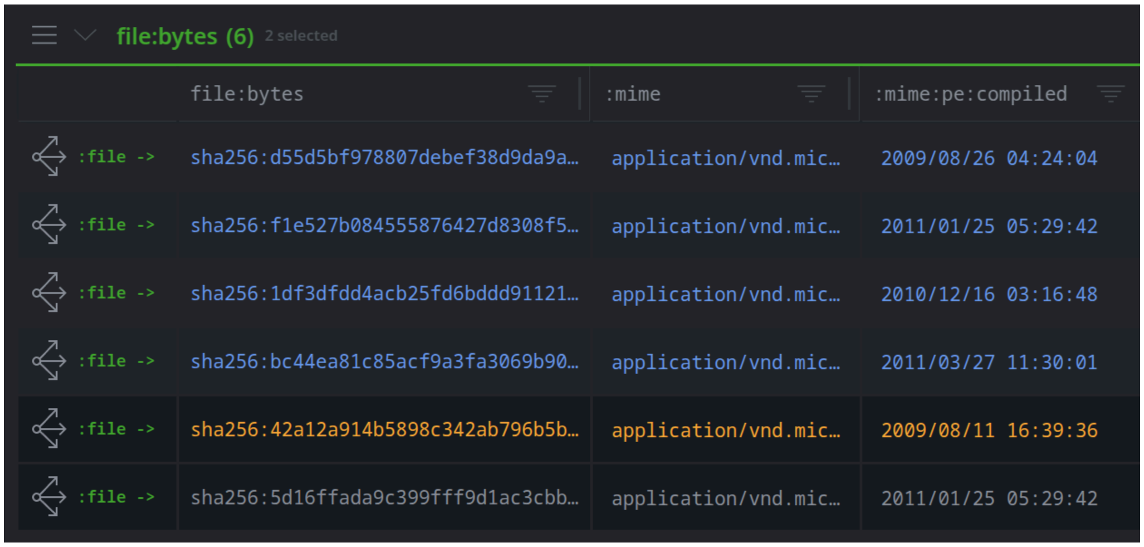

file:bytes (6)

file:bytes	:mime	:mime:pe:compiled	:mime:pe:imphash
sha256:d55d5bf978807debef3…	application/vnd…	2009/08/26 04:24:04	469387a37b85a92f0ea12f3…
sha256:f1e527b084555876427…	application/vnd…	2011/01/25 05:29:42	497181957fe081fed7f8110…
sha256:1df3dfdd4acb25fd6bd…	application/vnd…	2010/12/16 03:16:48	497181957fe081fed7f8110…
sha256:bc44ea81c85acf9a3fa…	application/vnd…	2011/03/27 11:30:01	25a95f3096565d09833c571…
sha256:42a12a914b5898c342a…	application/vnd…	2009/08/11 16:39:36	7a75b7d1e0076e41f8f57a5…
sha256:5d16ffada9c399fff9d…	application/vnd…	2011/01/25 05:29:42	497181957fe081fed7f8110…

  - Symantec associates **three** files with **Comment Crew.**
  - Mandiant associates **one** file with **APT1.**

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

- One file (in **orange**) is only tagged by VirusTotal, and one file has no tags. These are worth investigating further!



file:bytes (6) 2 selected

	file:bytes	:mime	:mime:pe:compiled
:file ->	sha256:d55d5bf978807debef38d9da9a…	application/vnd.mic…	2009/08/26 04:24:04
:file ->	sha256:f1e527b084555876427d8308f5…	application/vnd.mic…	2011/01/25 05:29:42
:file ->	sha256:1df3dfdd4acb25fd6bddd91121…	application/vnd.mic…	2010/12/16 03:16:48
:file ->	sha256:bc44ea81c85acf9a3fa3069b90…	application/vnd.mic…	2011/03/27 11:30:01
:file ->	sha256:42a12a914b5898c342ab796b5b…	application/vnd.mic…	2009/08/11 16:39:36
:file ->	sha256:5d16ffada9c399fff9d1ac3cbb…	application/vnd.mic…	2011/01/25 05:29:42