Static Analysis – Techniques and Tooling

Static analysis covers everything that can obtain information from a sample without actually loading the program into executable memory space and observing its behavior.

Technical requirements

The technical requirements for this chapter are as follows:

• FLARE VM

• An internet connection (connect only when required)

- Tools - **Get-FileHash, Get-ChildItem, VirusTotal, ssdeep, filetype.exe, strings**

• .zip files containing tools and malware samples from the desktop or downloaded from the below link:

https://github.com/ PacktPublishing/Malware-Analysis-Techniques

Text book : Malware Analysis Techniques Tricks for the triage of adversarial software Dylan Barker, 2021

Hashing

- A hashing algorithm is a one-way function that generates a unique checksum for every file, much like a fingerprint of the file.
- Hashing algorithms and their corresponding bits:

Algorithm	Output Bits	Broken
MD5	128	Yes
SHA1	160	Yes
SHA256	256	No
SHA512	512	No

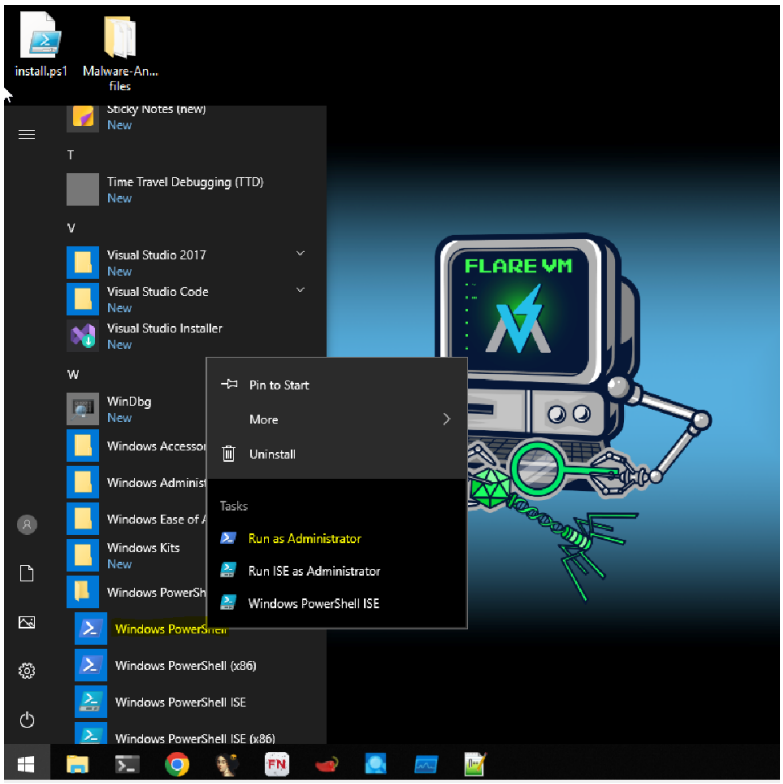
- **Collision** is an occurrence where two different files have identical hashes. When a collision occurs, a hashing algorithm is considered broken and no longer reliable. Examples of such algorithms include MD5 and SHA1.

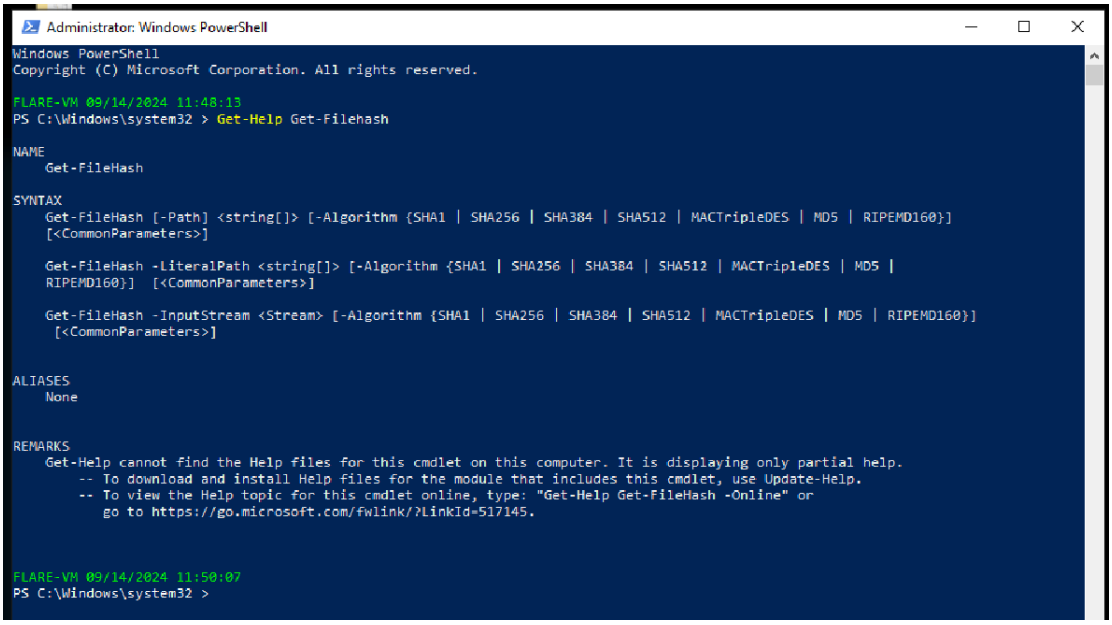
➢ **Obtaining file hashes**

- Tool built-in into Windows PowerShell - **Get-FileHash** — gets the hash of the file it is provided.
- **Get-ChildItem** and piping the output to Get-FileHash is a great way to get the hashes of files in bulk.
- Modes in Get-ChildItem

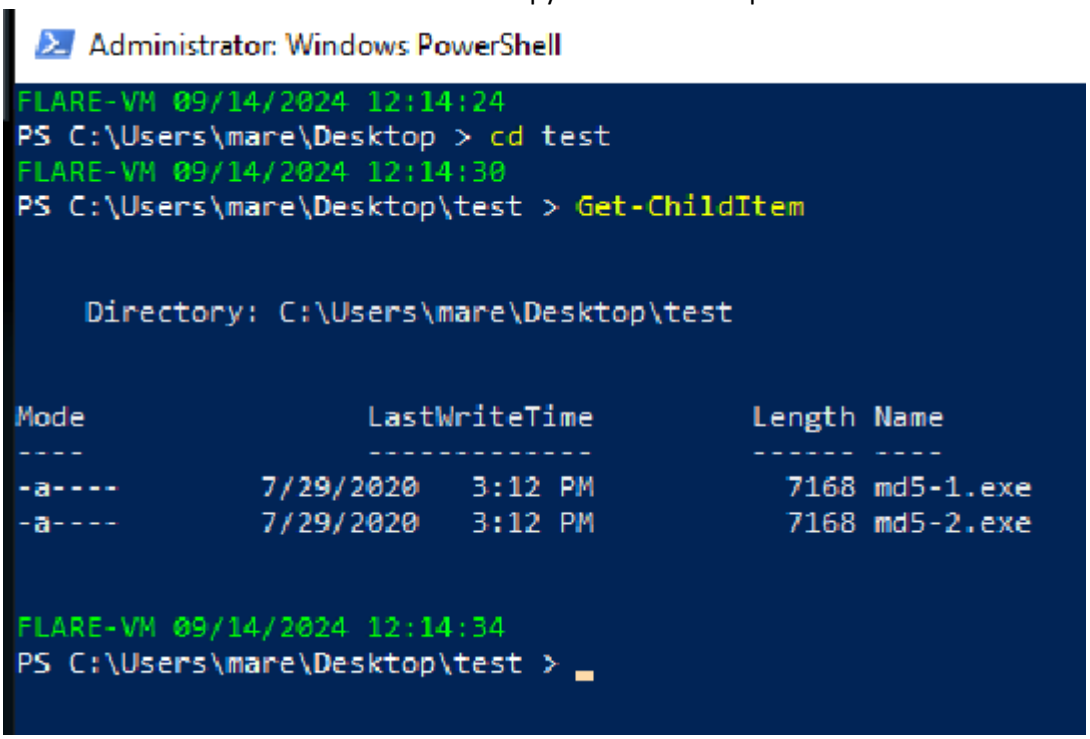
```
d - Directory
a - Archive
r - Read-only
h - Hidden
s - System
l - Reparse point, symlink, etc.
```

- Usage of the cmdlet by typing Get-Help Get-FileHash

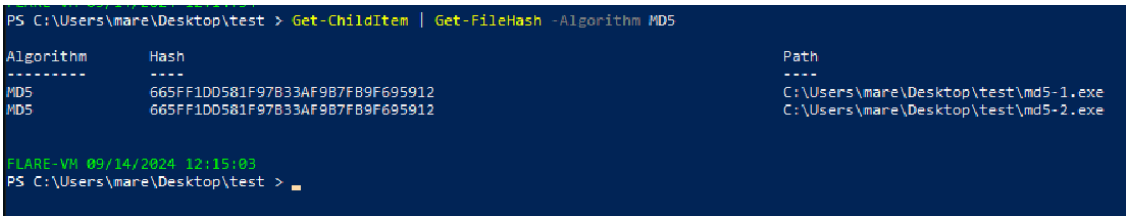




- Use the files - md5-1.exe and md5-2.exe. Copy them to Desktop. Use the tool Get-ChildItem



- Find the hash for both the files



- **Observation:** the files have the same size, MD5 hash

- Reconfirm with a different hash algorithm

```
PS C:\Users\mare\Desktop\test > Get-ChildItem | Get-FileHash -Algorithm SHA256

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          E16A3E7BEA60AB2AA1E49E31199791648C58B14D1691935F25F3BD4E94F2F34B       C:\Users\mare\Desktop\test\md5-1.exe
SHA256          84AF18CFD067DF107B790EDDE3DBD23A0379F8FBBD1913AB0CEA74C4378F4569       C:\Users\mare\Desktop\test\md5-2.exe


FLARE-VM 09/14/2024 12:17:32
PS C:\Users\mare\Desktop\test >
```

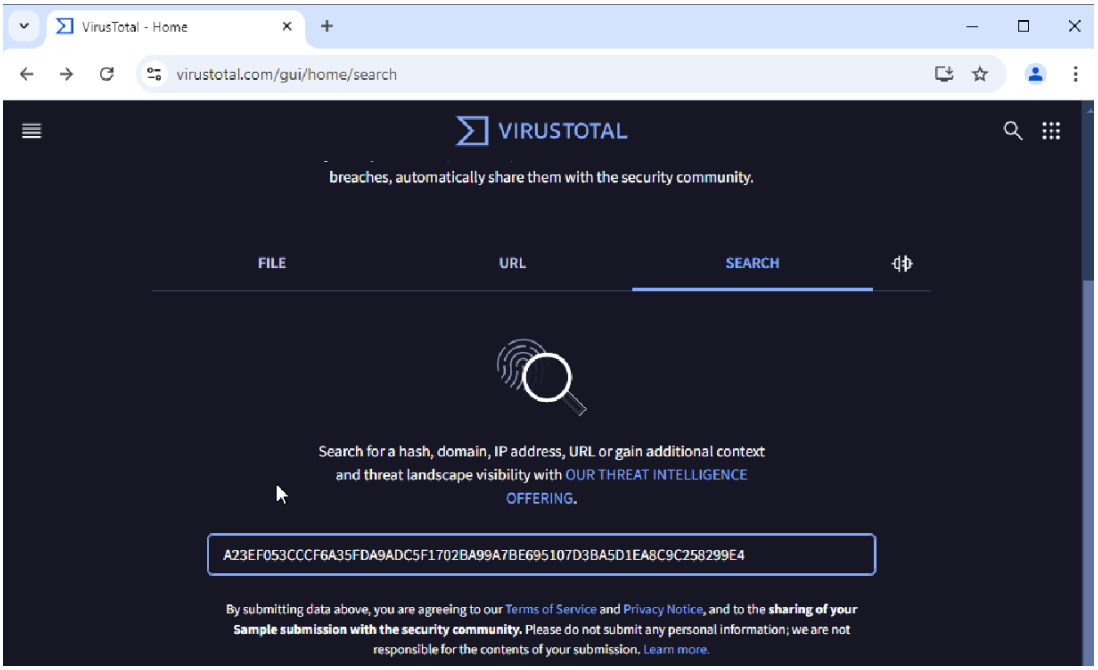
- **Observation:** SHA256 hashes differ! This indicates without a doubt that these files, while the same size and with the same MD5 hash, are not the same file. The importance of choosing a strong one-way hashing algorithm is demonstrated.
- **VirusTotal** - scanning engine that scans possible malware samples against several antivirus (AV) engines and reports their findings. https://virustotal.com/
- Use another sample— 8888888.png. file from chapter 2 folder.

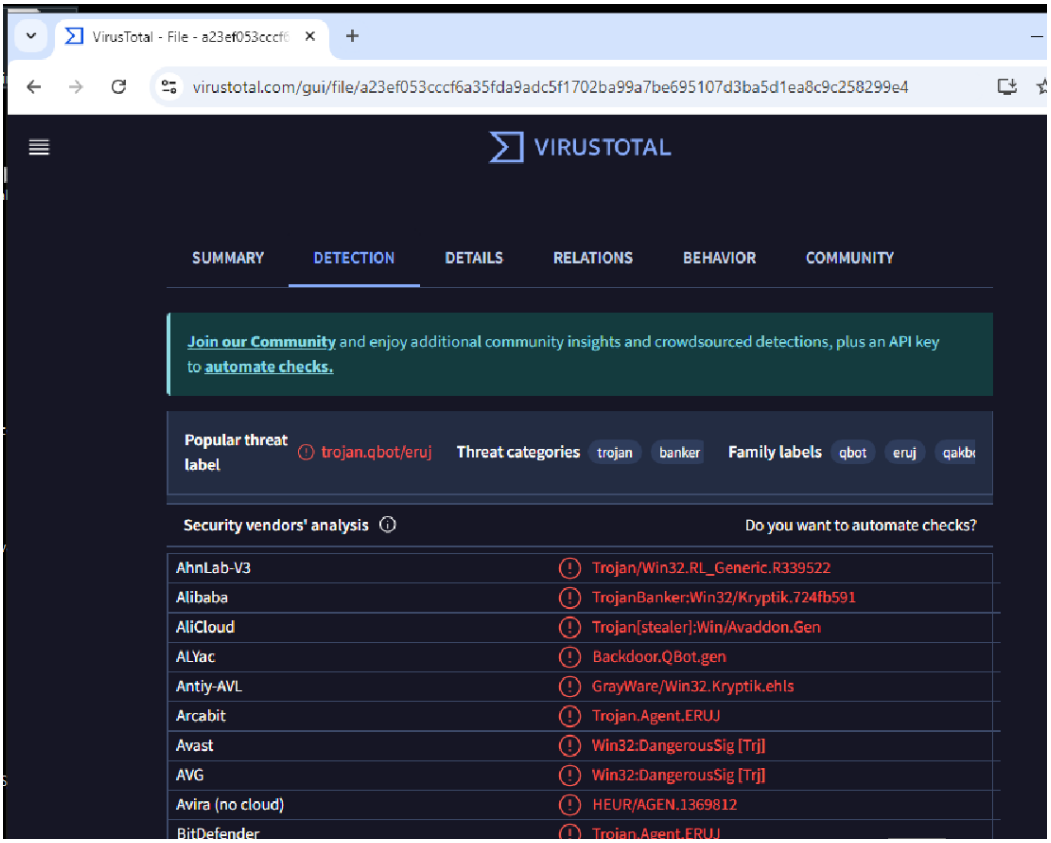
> **Warning!**
>
> 888888.png is live malware—a sample of the **Qakbot** (**QBot**) banking Trojan threat! Handle this sample with care!

- Find the hash value using the tools Get-ChildItem and Get-Filehash

```
PS C:\Users\mare\Desktop\test > Get-ChildItem | Get-FileHash -Algorithm SHA256

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          A23EF053CCCF6A35FDA9ADC5F1702BA99A7BE695107D3BA5D1EA8C9C258299E4       C:\Users\mare\Desktop\test\8888888.png
SHA256          E16A3E7BEA60AB2AA1E49E31199791648C58B14D1691935F25F3BD4E94F2F34B       C:\Users\mare\Desktop\test\md5-1.exe
SHA256          84AF18CFD067DF107B790EDDE3DBD23A0379F8FBBD1913AB0CEA74C4378F4569       C:\Users\mare\Desktop\test\md5-2.exe


FLARE-VM 09/14/2024 12:33:51
PS C:\Users\mare\Desktop\test >
```

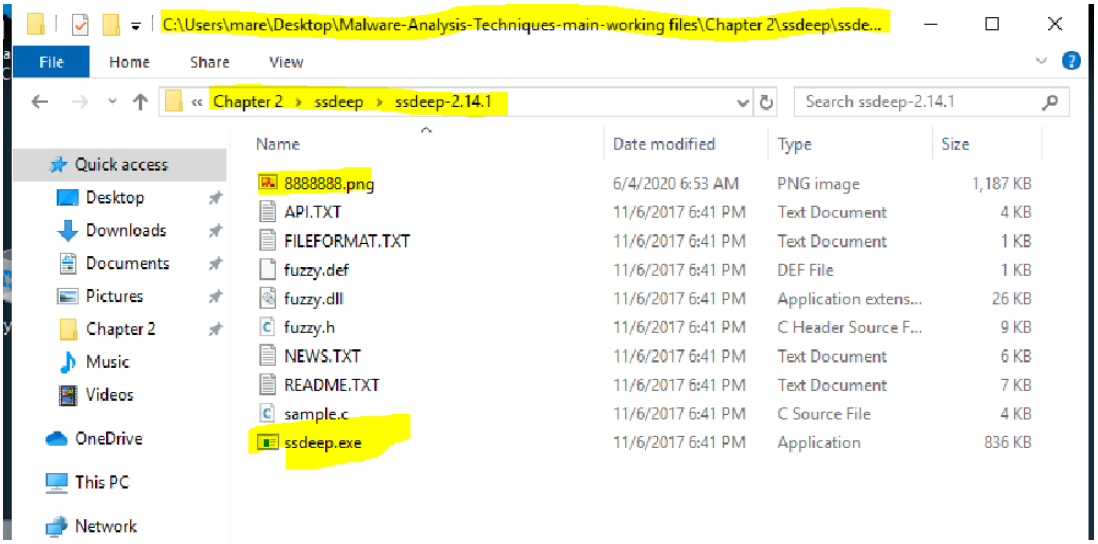
- copy the hash and paste in the VirusTotal web link to check possible match with several antivirus engines.



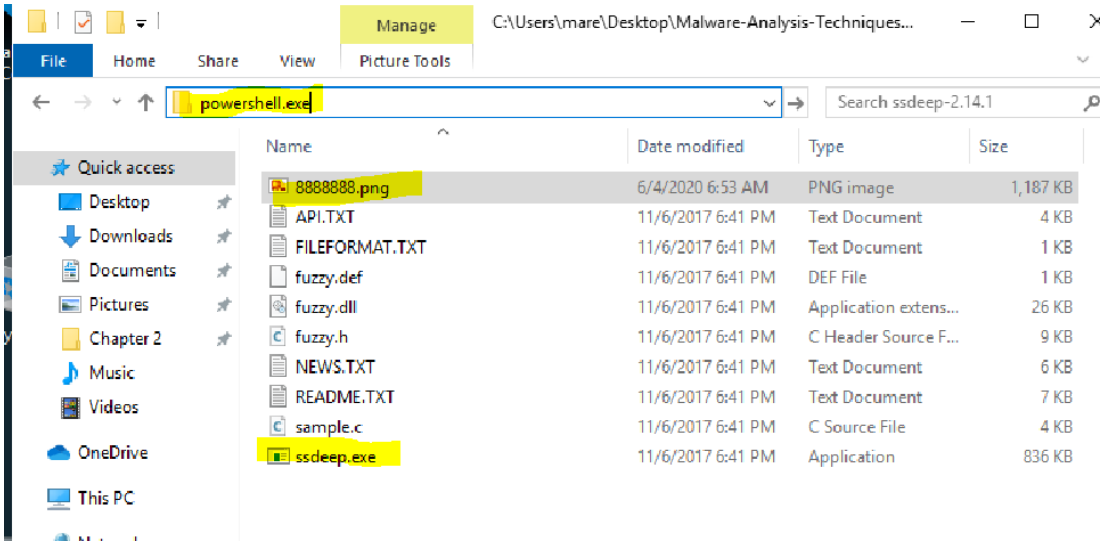


- if no matches found, then **hashbusting** technique (ensures each malware sample has a different static hash!) might be used.
- **fuzzy hashing** – handles hashbusting
- **ssdeep** is a fuzzy hashing algorithm that utilizes a similarity digest in order to create and output representations of files in the following format:

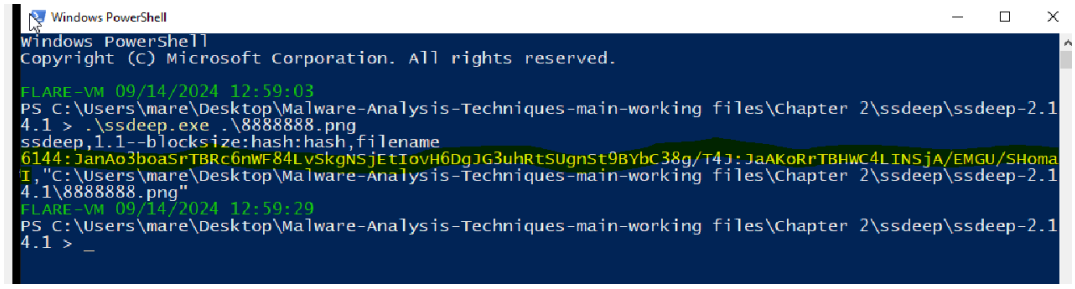
  chunksize:chunk:double_chunk
- Extract ssdeep.exe, copy and paste the file 8888888.png in the same folder as ssdeep.exe



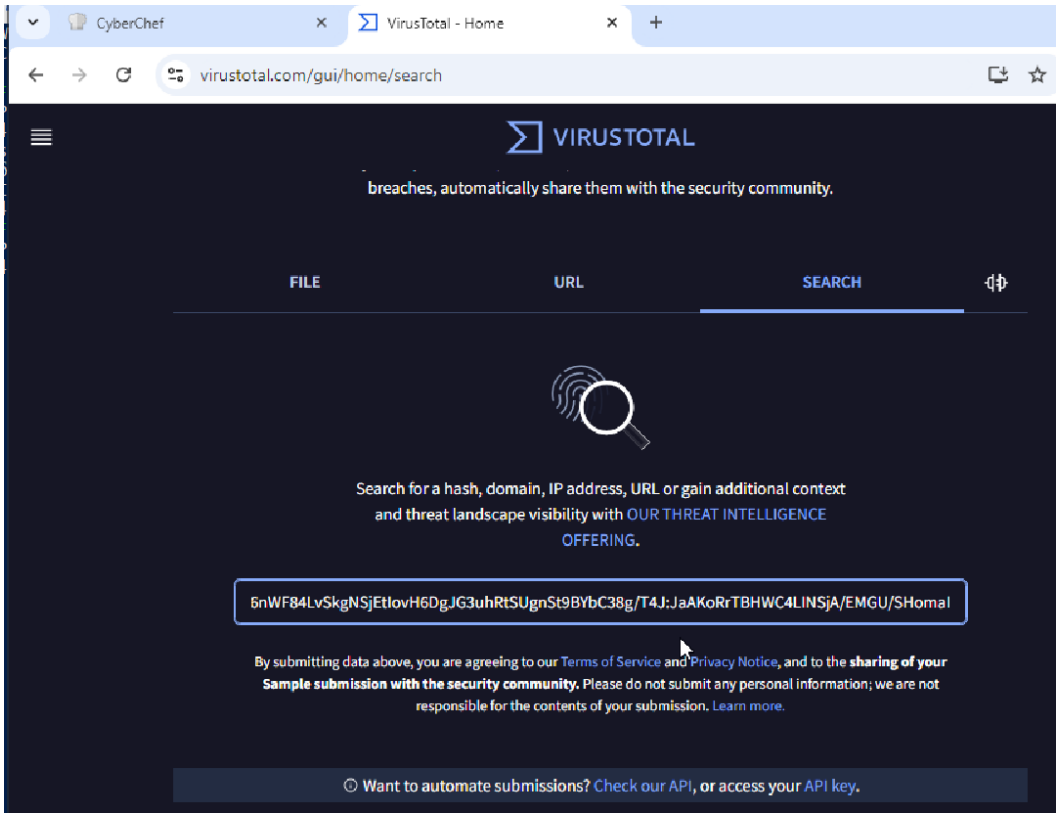
- Open the PowerShell window to the location where ssdeep file is available.

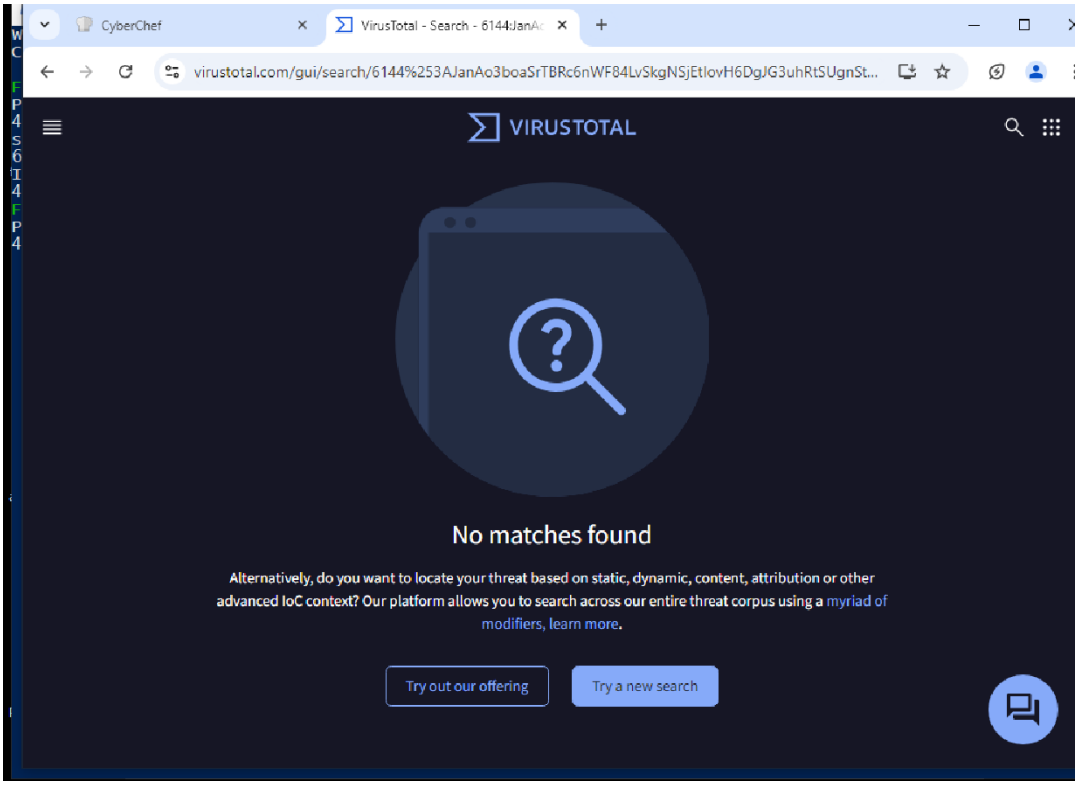


- Obtain ssdeep hash using: **.\ssdeep.exe .\8888888.png**

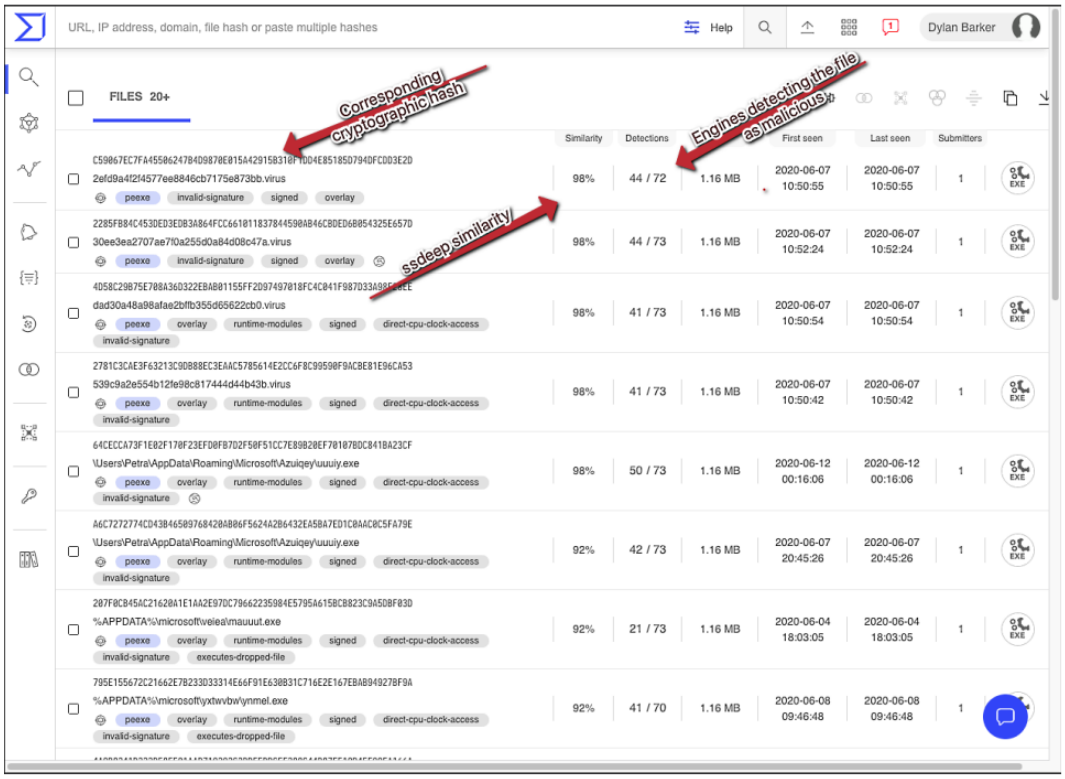


- **Launch VirusTotal.com and find the matches**

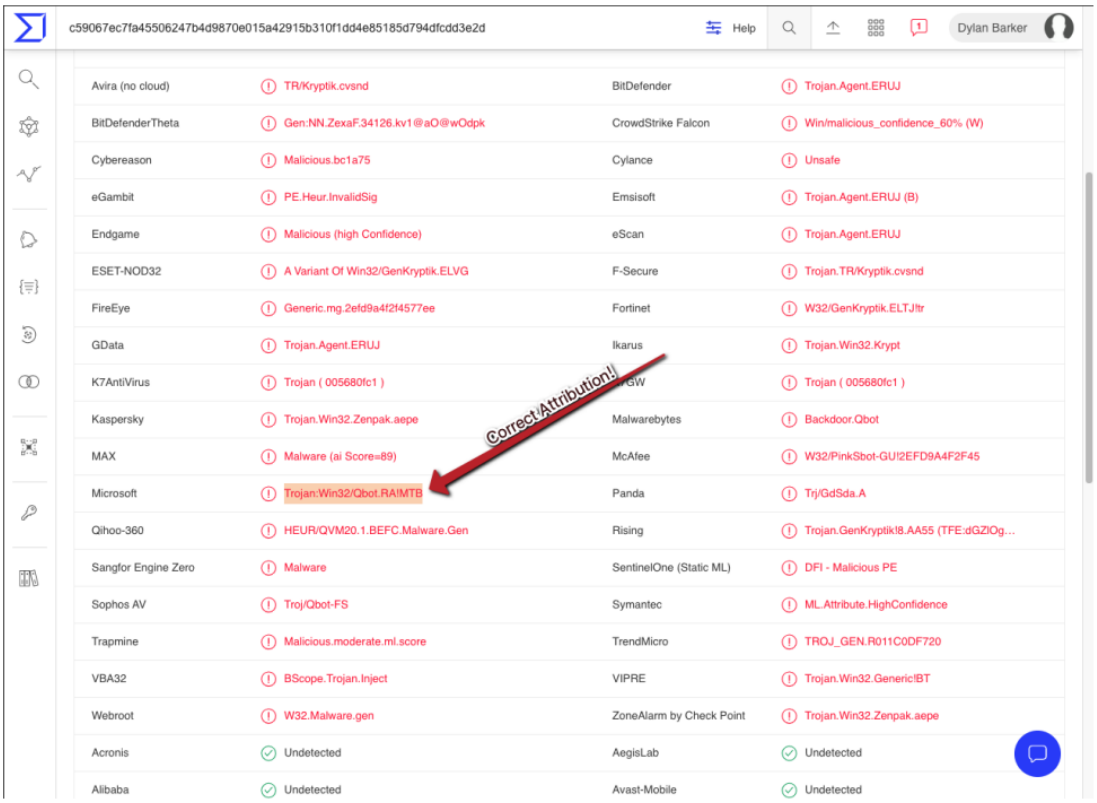




- ssdeep hashes in VirusTotal, requires an Enterprise membership.
- Observe the following screenshot and interpret.



- Clicking one of the highly similar cryptographic hashes will load the VirusTotal scan results for the sample



- **Malware serotyping** - Adversaries frequently change the extension of files, sometimes excluding it altogether and sometimes creating double extensions, such as notmalware.doc.exe
- Need a utility for testing file types like **filetype.exe –** a tool within FLARE

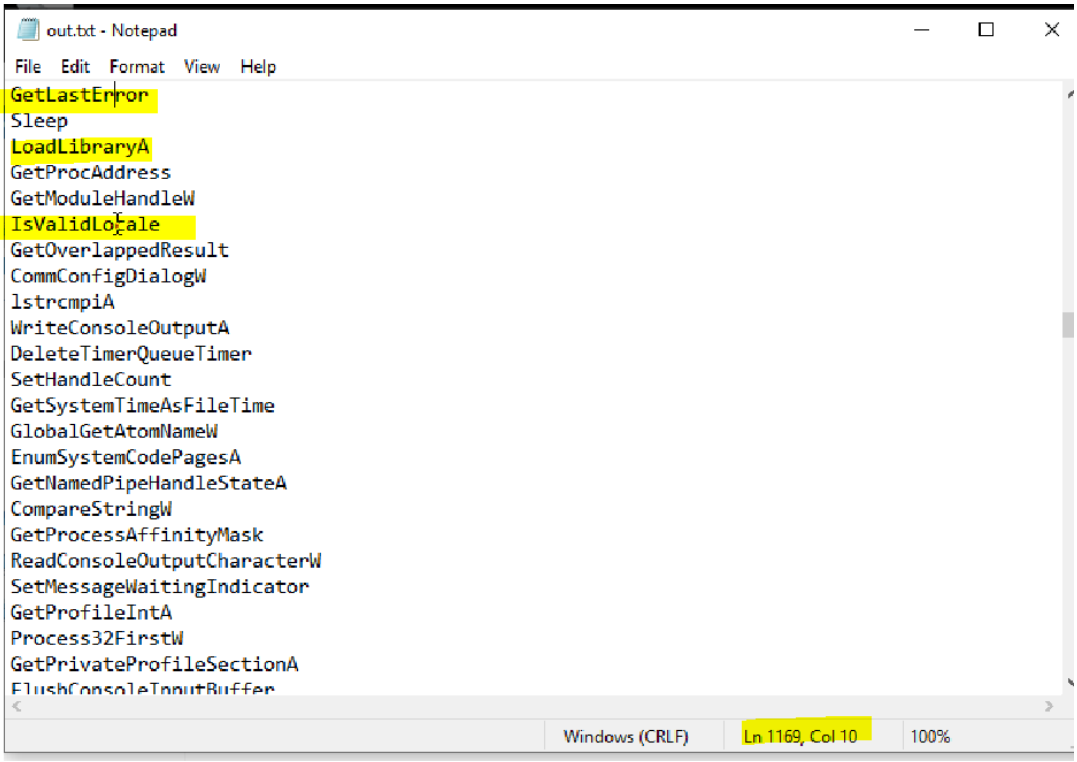
```
Windows PowerShell
FLARE-VM 09/14/2024 13:21:13
PS C:\Users\mare\Desktop\MAT-working files\Chapter 2 > .\filetype -i .\8888888.png
.\8888888.png (.exe) "Executable File"
FLARE-VM 09/14/2024 13:21:25
PS C:\Users\mare\Desktop\MAT-working files\Chapter 2 > _
```

- **Observation :** adversary has intentionally changed the file type from .exe to .png

- **Collecting strings**- When an executable is compiled, certain ASCII- or Unicode-encoded strings used during development may be included in the binary
- **strings** - a tool from Microsoft's Windows Sysinternals can be utilized to extract any strings located within the binary
- Launch the command prompt and type **strings -n 5 8888888. png > output.txt**
  -n, the minimum string length to return

```
C:\Windows\system32\cmd.exe

FLARE-VM Sat 09/14/2024 13:46:09.32
C:\Users\mare\Desktop\MAT-working files\Chapter 2>strings -n 5 8888888.png > out.txt

Strings v2.54 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2021 Mark Russinovich
Sysinternals - www.sysinternals.com


FLARE-VM Sat 09/14/2024 13:46:17.34
C:\Users\mare\Desktop\MAT-working files\Chapter 2>dir
 Volume in drive C has no label.
 Volume Serial Number is D42E-325A

 Directory of C:\Users\mare\Desktop\MAT-working files\Chapter 2

09/14/2024  01:45 PM    <DIR>          .
09/14/2024  01:45 PM    <DIR>          ..
06/04/2020  06:53 AM         1,214,992 8888888.png
02/27/2021  08:27 AM           184,320 Challenge_1.dll
07/18/2021  01:52 PM         3,723,264 Challenge_2.bin
09/03/2004  07:39 AM            45,056 filetype.exe
09/03/2004  07:29 AM             4,372 filetypes.dat
07/29/2020  03:12 PM             7,168 md5-1.exe
07/29/2020  03:12 PM             7,168 md5-2.exe
09/14/2024  01:46 PM            40,752 out.txt
09/14/2024  12:55 PM    <DIR>          ssdeep
07/18/2021  01:55 PM           331,210 ssdeep.zip
               9 File(s)      5,558,302 bytes
               3 Dir(s)  36,129,927,168 bytes free

FLARE-VM Sat 09/14/2024 13:46:35.15
C:\Users\mare\Desktop\MAT-working files\Chapter 2>
```

- There are several strings have been returned, including some of the **Windows application programming interface (API)** modules that are imported by this binary.



- We can gain some information on which executable was **backdoored** or what the binary is **masquerading as**! This may prove useful both in tracking the operations of the campaign and tracking **indicators of compromise (IOCs)** for internal outbreaks.