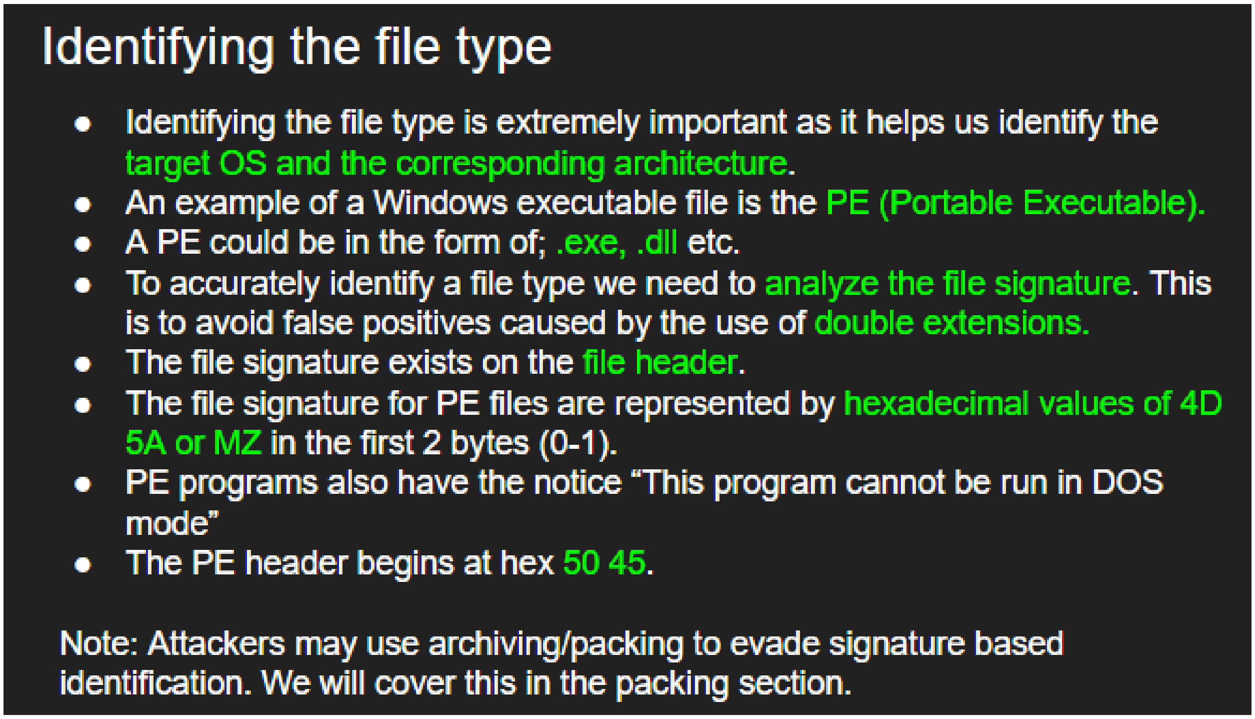
MALWARE ANALYSIS

FILE TYPE IDENTIFICATION:

https://www.youtube.com/watch?v=idcvzyibrag&list=PLBf0hzazHTGMSlOI2HZGc08ePwut6A2Io&index=8



Identifying the file type

- Identifying the file type is extremely important as it helps us identify the target OS and the corresponding architecture.
- An example of a Windows executable file is the PE (Portable Executable).
- A PE could be in the form of; .exe, .dll etc.
- To accurately identify a file type we need to analyze the file signature. This is to avoid false positives caused by the use of double extensions.
- The file signature exists on the file header.
- The file signature for PE files are represented by hexadecimal values of 4D 5A or MZ in the first 2 bytes (0-1).
- PE programs also have the notice "This program cannot be run in DOS mode"
- The PE header begins at hex 50 45.

Note: Attackers may use archiving/packing to evade signature based identification. We will cover this in the packing section.

Hackers try to change the extension. Ex: change .exe to doc file or file.exe.doc

Therefore, file signature is important.

Downloads required: (other than YARA)

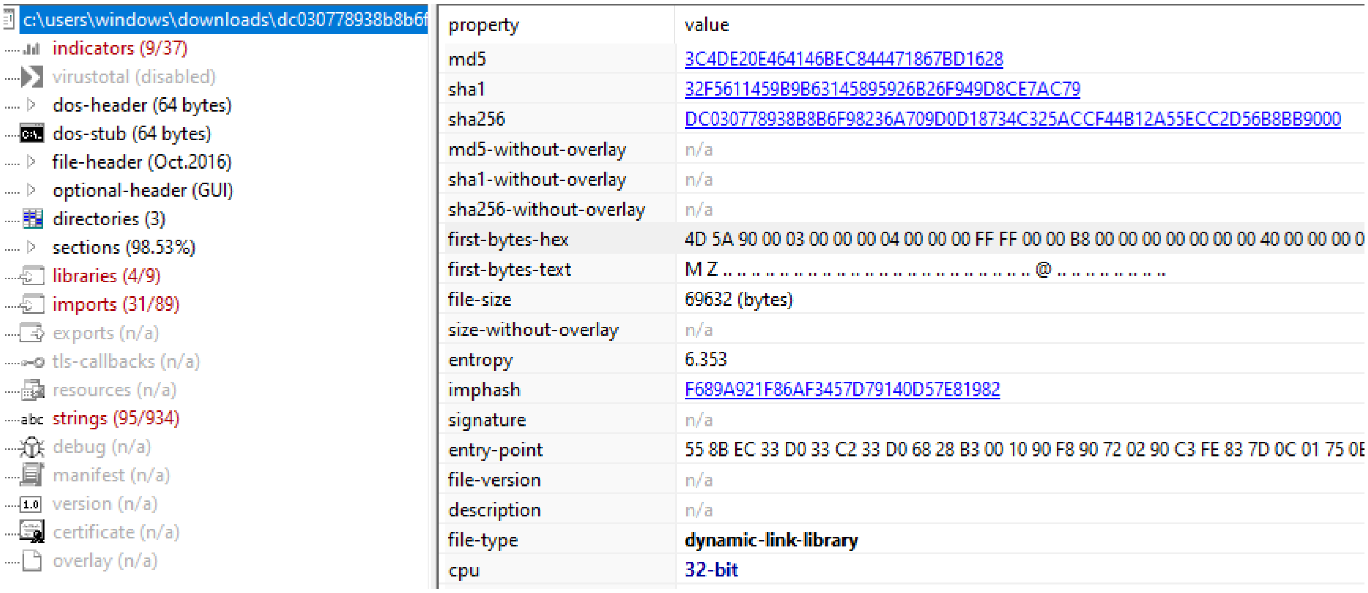
Malware Sample download: https://s3.eu-central-1.amazonaws.com...

It is a generic password stealer/credential harvester.

PEStudio: https://www.winitor.com/features

Steps to setup the file:

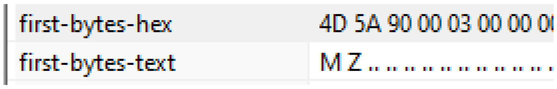
- Extract the file
- It will ask for password: "infected".
- Notice that it doesn't have an extension but it doesn't mean it can't be an executable.
- Drag and drop in pestudio.



c:\users\windows\downloads\dc030778938b8b6f
- indicators (9/37)
- virustotal (disabled)
- dos-header (64 bytes)
- dos-stub (64 bytes)
- file-header (Oct.2016)
- optional-header (GUI)
- directories (3)
- sections (98.53%)
- libraries (4/9)
- imports (31/89)
- exports (n/a)
- tls-callbacks (n/a)
- resources (n/a)
- strings (95/934)
- debug (n/a)
- manifest (n/a)
- version (n/a)
- certificate (n/a)
- overlay (n/a)

property	value
md5	3C4DE20E464146BEC844471867BD1628
sha1	32F5611459B9B63145895926B26F949D8CE7AC79
sha256	DC030778938B8B6F98236A709D0D18734C325ACCF44B12A55ECC2D56B8BB9000
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 0
first-bytes-text	M Z .. .. @ .. ..
file-size	69632 (bytes)
size-without-overlay	n/a
entropy	6.353
imphash	F689A921F86AF3457D79140D57E81982
signature	n/a
entry-point	55 8B EC 33 D0 33 C2 33 D0 68 28 B3 00 10 90 F8 90 72 02 90 C3 FE 83 7D 0C 01 75 0E
file-version	n/a
description	n/a
file-type	**dynamic-link-library**
cpu	**32-bit**

You can see first bytes as 4D 5A i.e M Z which tells us it is a PE file.

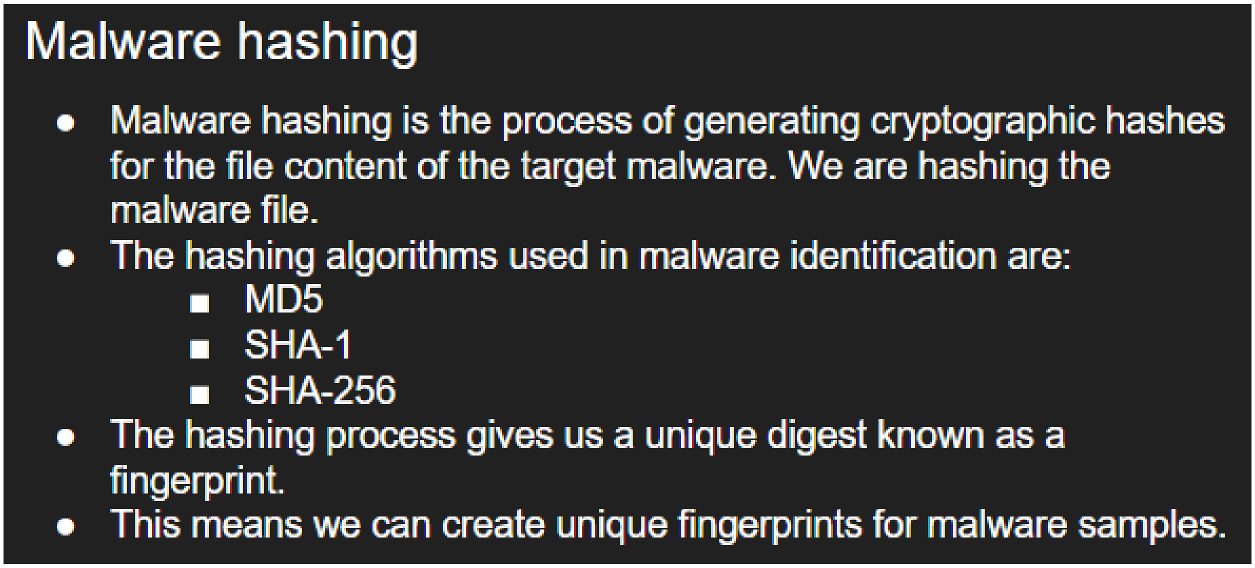


first-bytes-hex	4D 5A 90 00 03 00 00 00
first-bytes-text	M Z .. ..

File type is DLL

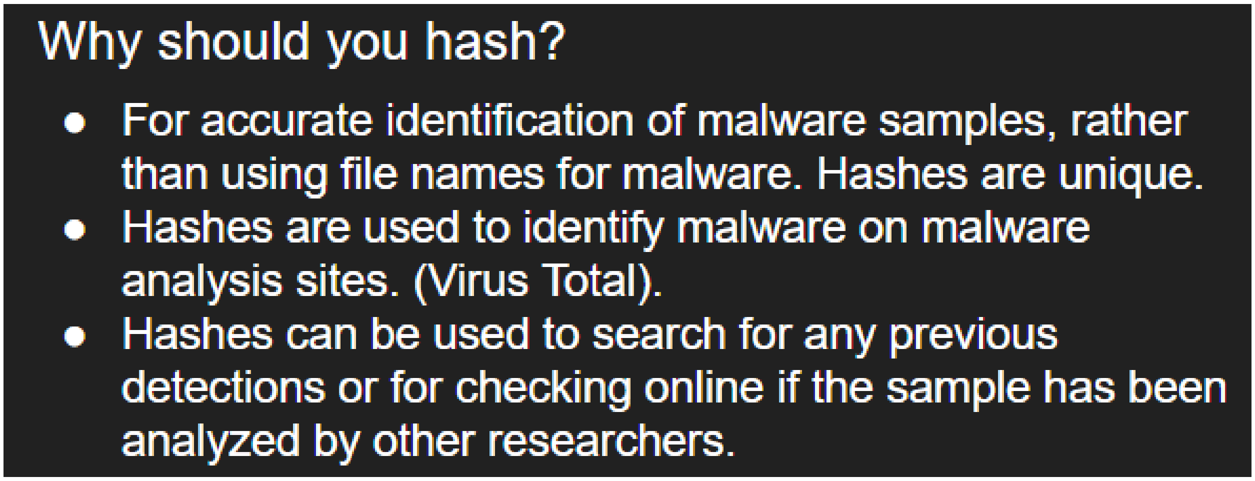
MALWARE HASHES AND VIRUS TOTAL:

https://www.youtube.com/watch?v=-Z0d6q73Lsg&list=PLBf0hzazHTGMSlOI2HZGc08ePwut6A2Io&index=9



Malware hashing

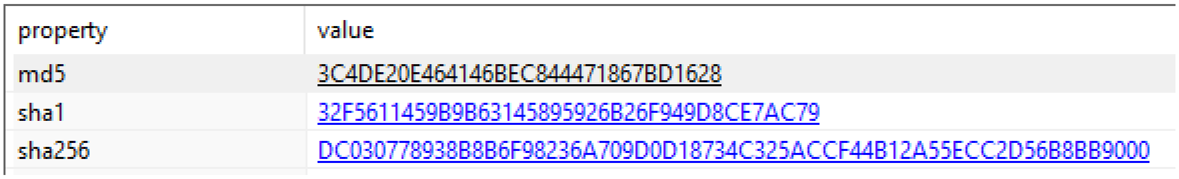
- Malware hashing is the process of generating cryptographic hashes for the file content of the target malware. We are hashing the malware file.
- The hashing algorithms used in malware identification are:
  - MD5
  - SHA-1
  - SHA-256
- The hashing process gives us a unique digest known as a fingerprint.
- This means we can create unique fingerprints for malware samples.



Why should you hash?

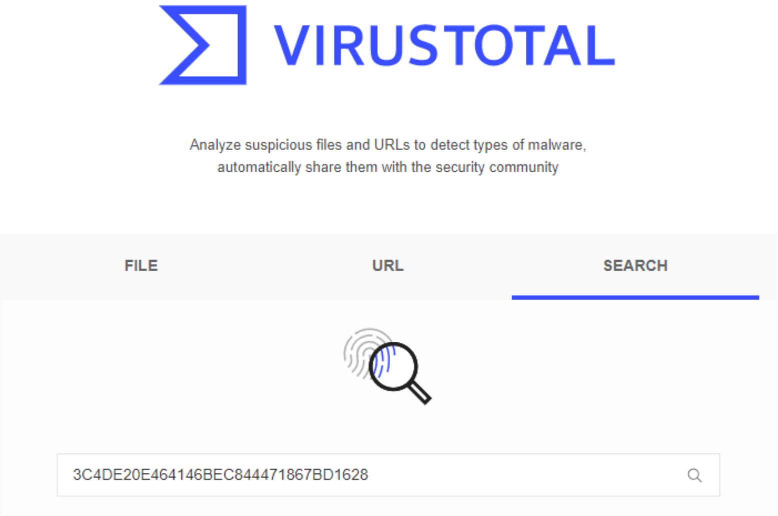
- For accurate identification of malware samples, rather than using file names for malware. Hashes are unique.
- Hashes are used to identify malware on malware analysis sites. (Virus Total).
- Hashes can be used to search for any previous detections or for checking online if the sample has been analyzed by other researchers.

You can see the different hash values in PEstudio.



property	value
md5	3C4DE20E464146BEC844471867BD1628
sha1	32F5611459B9B63145895926B26F949D8CE7AC79
sha256	DC030778938B8B6F98236A709D0D18734C325ACCF44B12A55ECC2D56B8BB9000

Copy md5 hash and search it on virus total website

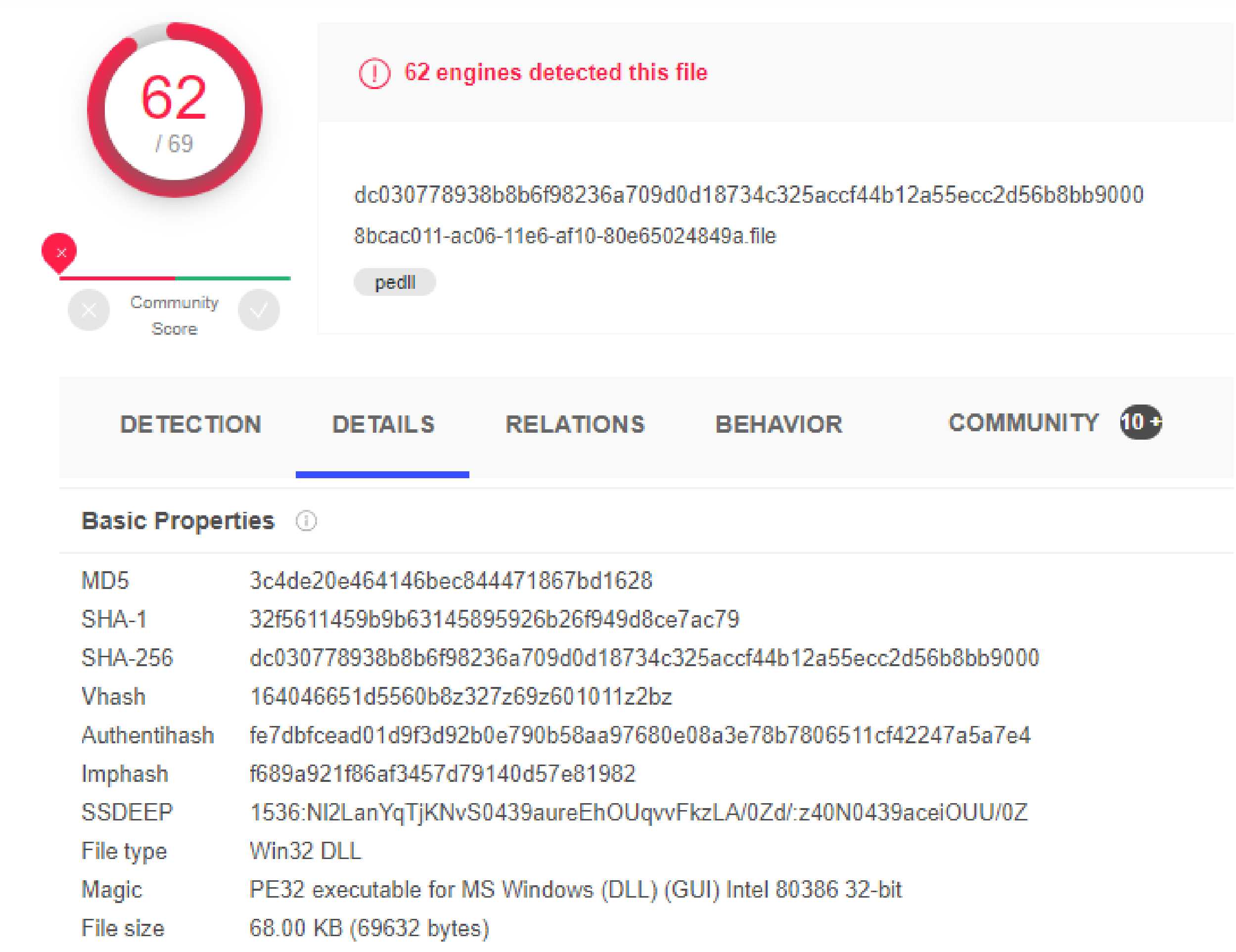


Analyze suspicious files and URLs to detect types of malware, automatically share them with the security community

3C4DE20E464146BEC844471867BD1628

62/69 engines have detected this malware before.

Select details and you can see the file type, other hash values ,its history, various names it goes by etc.



62 engines detected this file

dc030778938b8b6f98236a709d0d18734c325accf44b12a55ecc2d56b8bb9000

8bcac011-ac06-11e6-af10-80e65024849a.file

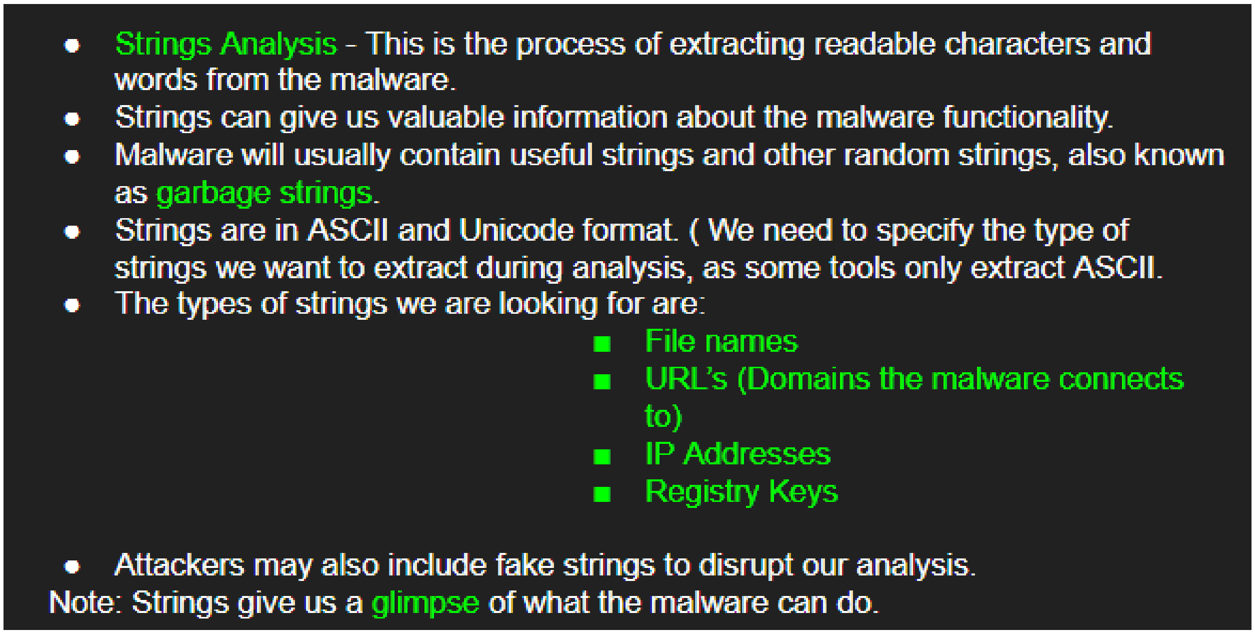
pedll

Basic Properties

MD5	3c4de20e464146bec844471867bd1628
SHA-1	32f5611459b9b63145895926b26f949d8ce7ac79
SHA-256	dc030778938b8b6f98236a709d0d18734c325accf44b12a55ecc2d56b8bb9000
Vhash	164046651d5560b8z327z69z601011z2bz
Authentihash	fe7dbfcead01d9f3d92b0e790b58aa97680e08a3e78b7806511cf42247a5a7e4
Imphash	f689a921f86af3457d79140d57e81982
SSDEEP	1536:NI2LanYqTjKNvS0439aureEhOUqvvFkzLA/0Zd/:z40N0439aceiOUU/0Z
File type	Win32 DLL
Magic	PE32 executable for MS Windows (DLL) (GUI) Intel 80386 32-bit
File size	68.00 KB (69632 bytes)

ANALYZING STRINGS

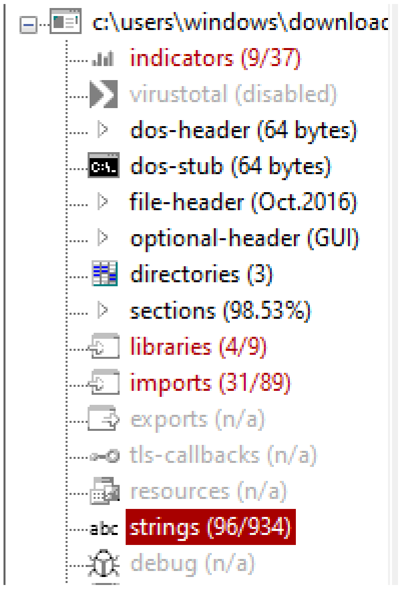
https://www.youtube.com/watch?v=V3_vc7BO9lU&list=PLBf0hzazHTGMSlOI2HZGc08ePwut6A2Io&index=10



- Strings Analysis - This is the process of extracting readable characters and words from the malware.
- Strings can give us valuable information about the malware functionality.
- Malware will usually contain useful strings and other random strings, also known as garbage strings.
- Strings are in ASCII and Unicode format. (We need to specify the type of strings we want to extract during analysis, as some tools only extract ASCII.
- The types of strings we are looking for are:
  - File names
  - URL's (Domains the malware connects to)
  - IP Addresses
  - Registry Keys
- Attackers may also include fake strings to disrupt our analysis.

Note: Strings give us a glimpse of what the malware can do.

Select strings section in PEStudio to analyze them and find strings that might identify the malware family.

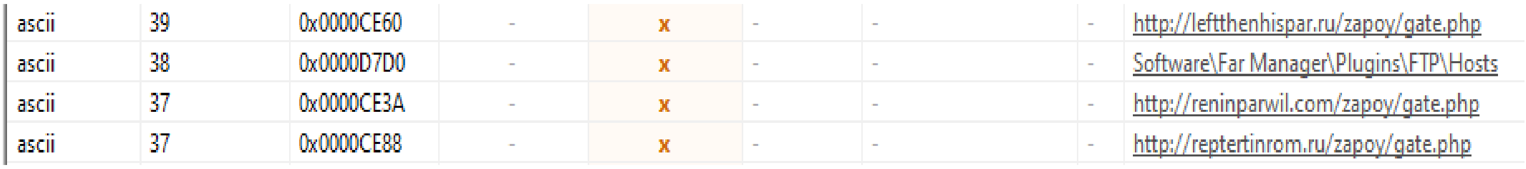


c:\users\windows\downloac
indicators (9/37)
virustotal (disabled)
dos-header (64 bytes)
dos-stub (64 bytes)
file-header (Oct.2016)
optional-header (GUI)
directories (3)
sections (98.53%)
libraries (4/9)
imports (31/89)
exports (n/a)
tls-callbacks (n/a)
resources (n/a)
strings (96/934)
debug (n/a)

You can observe a number of things.

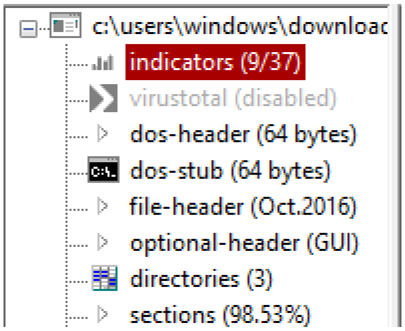
A POST method called meaning that the hacker might be sending information. We can see registry key strings too.

Notice the three url strings. They might be the Russian command and control centers from where the attack is controlled or where all the credentials are sent back to.

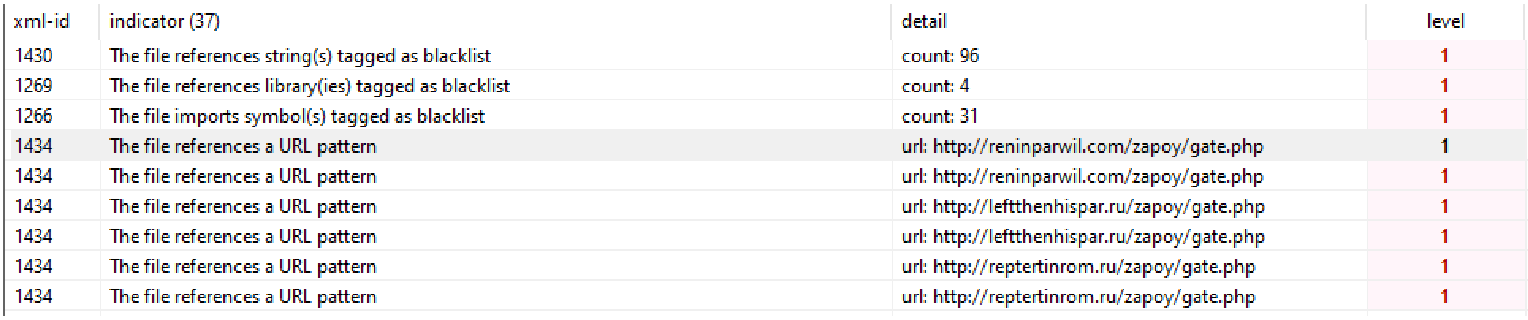


ascii	39	0x0000CE60	-	x	-	-	-	http://leftthenhispar.ru/zapoy/gate.php
ascii	38	0x0000D7D0	-	x	-	-	-	Software\Far Manager\Plugins\FTP\Hosts
ascii	37	0x0000CE3A	-	x	-	-	-	http://reninparwil.com/zapoy/gate.php
ascii	37	0x0000CE88	-	x	-	-	-	http://reptertinrom.ru/zapoy/gate.php

For more help checkout the indicators section.



c:\users\windows\downloac
indicators (9/37)
virustotal (disabled)
dos-header (64 bytes)
dos-stub (64 bytes)
file-header (Oct.2016)
optional-header (GUI)
directories (3)
sections (98.53%)



xml-id	indicator (37)	detail	level
1430	The file references string(s) tagged as blacklist	count: 96	1
1269	The file references library(ies) tagged as blacklist	count: 4	1
1266	The file imports symbol(s) tagged as blacklist	count: 31	1
1434	The file references a URL pattern	url: http://reninparwil.com/zapoy/gate.php	1
1434	The file references a URL pattern	url: http://reninparwil.com/zapoy/gate.php	1
1434	The file references a URL pattern	url: http://leftthenhispar.ru/zapoy/gate.php	1
1434	The file references a URL pattern	url: http://leftthenhispar.ru/zapoy/gate.php	1
1434	The file references a URL pattern	url: http://reptertinrom.ru/zapoy/gate.php	1
1434	The file references a URL pattern	url: http://reptertinrom.ru/zapoy/gate.php	1

url: http://reninparwil.com/zapoy/gate.php

url: http://leftthenhispar.ru/zapoy/gate.php

url: http://reptertinrom.ru/zapoy/gate.php

CREATING YARA RULE:

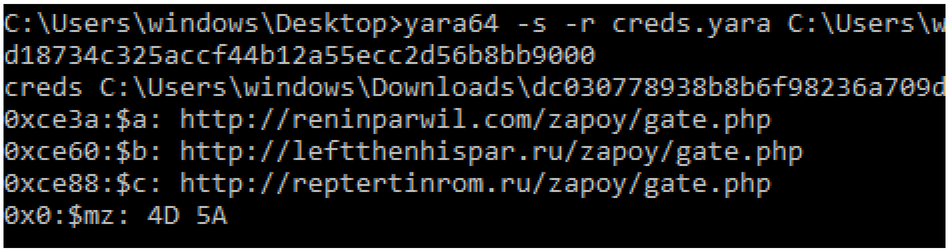
https://www.youtube.com/watch?v=35Exd9GrR5I&list=PLBf0hzazHTGMSlOI2HZGc08ePwut6A2Io&index=16

Why use yara rules?

Hashing is not accurate because any change the hacker makes, changes the hash value too (even though the functionality remains same). Yara rules are powerful because if the hacker uses the same functionality, yara can detect it.

So we will write a yara rule based on the C&C center urls which we identified just above. It will also identify a PE.

```
rule creds
{
meta:
      description = "Simple YARA rule to detect Command and control centers"
      date="12th June 2020"
strings:
      $a = "http://reninparwil.com/zapoy/gate.php"
      $b = "http://leftthenhispar.ru/zapoy/gate.php"
      $c = “http://reptertinrom.ru/zapoy/gate.php”
      $mz = {4D 5A}
condition:
      ($a or $b or $c)
}
```



```
C:\Users\windows\Desktop>yara64 -s -r creds.yara C:\Users\w
d18734c325accf44b12a55ecc2d56b8bb9000
creds C:\Users\windows\Downloads\dc030778938b8b6f98236a709d
0xce3a:$a: http://reninparwil.com/zapoy/gate.php
0xce60:$b: http://leftthenhispar.ru/zapoy/gate.php
0xce88:$c: http://reptertinrom.ru/zapoy/gate.php
0x0:$mz: 4D 5A
```

Virus companies use yara rules to identify malware (not as simple as this one). Now, when you download a file, it can be scanned to check if it’s a PE and if it interacts with those malicious urls.