Username: Palm Beach State College IP Holder **Book:** Kali Linux – Assuring Security by Penetration Testing. No part of any chapter or book may be reproduced or transmitted in any form by any means without the prior written permission for reprints and excerpts from the publisher of the book or chapter. Redistribution or other use that violates the fair use privilege under U.S. copyright laws (see 17 USC107) or that otherwise violates these Terms of Service is strictly prohibited. Violators will be prosecuted to the full extent of U.S. Federal and Massachusetts laws.

OS fingerprinting

After we know that the target machine is a live, we can then find out the operating system used by the target machine. This method is commonly known as **Operating System** (**OS**) fingerprinting. There are two methods of doing OS fingerprinting: **active** and **passive**.

In the active method, the tool sends network packets to the target machine and then determines the operating system of the target machine based on the analysis done on the response it has received. The advantage of this method is that the fingerprinting process is fast. However, the disadvantage is that the target machine may notice our attempt to get its operating system's information.

To overcome the active method's disadvantage, there exists a passive method of OS fingerprinting. This method was pioneered by Michal Zalewsky when he released a tool called `p0f`. The disadvantage of the passive method is that the process will be slower than the active method.

In this section, we will describe a couple of tools that can be used for OS fingerprinting.

p0f

The `p0f` tool is used to fingerprint an operating system passively. It can be used to identify an operating system on the following machines:

- Machines that connect to your box (SYN mode; this is the default mode)
- Machines you connect to (SYN+ACK mode)
- Machines you cannot connect to (RST+ mode)
- Machines whose communications you can observe

The `p0f` tool works by analyzing the TCP packets sent during the network activities. Then, it gathers the statistics of special packets that are not standardized by default by any corporations. An example is that the Linux kernel uses a 64-byte ping datagram, whereas the Windows operating system uses a 32-byte ping datagram; or the **Time To Live** (**TTL**) value. For Windows, the TTL value is 128, while for Linux this TTL value varies between the Linux distributions. These information are then used by `p0f` to determine the remote machine's operating system.

> **Note**
>
> When using the `p0f` tool included with Kali Linux, we were not able to fingerprint the operating system on a remote machine. We figured out that the `p0f` tool has not updated its fingerprint database. Unfortunately, we couldn't find the latest version of the fingerprint database. So, we used `p0f` v3 (Version 3.06b) instead. To use this version of `p0f`, just download the TARBALL file from http://lcamtuf.coredump.cx/p0f3/releases/p0f-3.06b.tgz and compile the code by running the **build.sh** script. By default, the fingerprint database file (`p0f.fp`) location is in the current directory. If you want to change the location, for example, if you want to change the location to `/etc/p0f/p0f.fp`, you need to change this in the `config.h` file and recompile `p0f`. If you don't change the location, you may need to use the `-f` option to define the fingerprint database file location.

To access `p0f`, open a console and type `p0f -h`. This will display its usage and options' description.

Let's use `p0f` to identify the operating system used in a remote machine we are connecting to. Just type the following command in your console:

```
p0f -f /etc/p0f/p0f.fp -o p0f.log
```

This will read the fingerprint database from the `/etc/p0f/p0f.fp` file and save the log information to the `p0f.log` file. It will then display the following information:

```
--- p0f 3.06b by Michal Zalewski <lcamtuf@coredump.cx> ---

[+] Closed 1 file descriptor.
[+] Loaded 314 signatures from '/etc/p0f/p0f.fp'.
[+] Intercepting traffic on default interface 'eth0'.
[+] Default packet filtering configured [+VLAN].
[+] Log file 'p0f.log' opened for writing.
```

```
[+] Entered main event loop.
```

Next, you need to generate network activities involving a TCP connection, such as browsing to the remote machine or letting the remote machine to connect to your machine.

If `p0f` has successfully fingerprinted the operating system, you will see information of the remote machine's operating system in the console and in the logfile (`p0f.log`).

Following is the information displayed to the console:

```
.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (syn) ]-
|
| client   = 192.168.56.101/42819
| os       = Linux 3.x
| dist     = 0
| params   = none
| raw_sig  = 4:64+0:0:1460:mss*10,7:mss,sok,ts,nop,ws:df,id+:0
|
`----

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (mtu) ]-
|
| client   = 192.168.56.101/42819
| link     = Ethernet or modem
| raw_mtu  = 1500
|
`----

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (syn+ack) ]-
|
| server   = 192.168.56.102/80
| os       = Linux 2.6.x
| dist     = 0
| params   = none
| raw_sig  = 4:64+0:0:1460:mss*4,5:mss,sok,ts,nop,ws:df:0
|
`----

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (mtu) ]-
|
| server   = 192.168.56.102/80
| link     = Ethernet or modem
| raw_mtu  = 1500
|
`----

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (http request) ]-
|
| client   = 192.168.56.101/42819
| app      = Firefox 10.x or newer
| lang     = English
| params   = none
| raw_sig  = 1:Host,User-Agent,Accept=[text/html,application
```

```
/xhtml+xml,application/xml;q=0.9,*/*;q=0.8],Accept-Language=[en-US,en;
q=0.5],Accept-Encoding=[gzip, deflate],Connection=[keep-alive]:Accept-
Charset,Keep-Alive:Mozilla/5.0 (X11; Linux x86_64; rv:18.0) Gecko/20100101
Firefox/18.0 Iceweasel/18.0.1
|
`----

.-[ 192.168.56.101/42819 -> 192.168.56.102/80 (http response) ]-
|
| server   = 192.168.56.102/80
| app      = Apache 2.x
| lang     = none
| params   = none
| raw_sig  = 1:Date,Server,X-Powered-By=[PHP/5.2.4-2ubuntu5.10],?Content-
Length,Keep-Alive=[timeout=15, max=100],Connection=[Keep-Alive],Content-
Type:Accept-Ranges:Apache/2.2.8 (Ubuntu) DAV/2
|
`----
```

The following screenshot shows the content of the logfile:

```
[2013/06/28 22:47:57] mod=syn|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=cli|os=Linux 3.x|dist=0|params=none|raw_
sig=4:64+0:0:1460:mss*10,7:mss,sok,ts,nop,ws:df,id+:0
[2013/06/28 22:47:57] mod=mtu|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=cli|link=Ethernet or modem|raw_mtu=1500
[2013/06/28 22:47:57] mod=syn+ack|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=srv|os=Linux 2.6.x|dist=0|params=non
e|raw_sig=4:64+0:0:1460:mss*4,5:mss,sok,ts,nop,ws:df:0
[2013/06/28 22:47:57] mod=mtu|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=srv|link=Ethernet or modem|raw_mtu=1500
[2013/06/28 22:47:57] mod=http request|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=cli|app=Firefox 10.x or newer|l
ang=English|params=none|raw_sig=1:Host,User-Agent,Accept=[text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8]
,Accept-Language=[en-US,en;q=0.5],Accept-Encoding=[gzip, deflate],Connection=[keep-alive]:Accept-Charset,Keep-Alive:Mozill
a/5.0 (X11; Linux x86_64; rv:18.0) Gecko/20100101 Firefox/18.0 Iceweasel/18.0.1
[2013/06/28 22:47:57] mod=http response|cli=192.168.56.101/42819|srv=192.168.56.102/80|subj=srv|app=Apache 2.x|lang=none|p
arams=none|raw_sig=1:Date,Server,X-Powered-By=[PHP/5.2.4-2ubuntu5.10],?Content-Length,Keep-Alive=[timeout=15, max=100],Con
nection=[Keep-Alive],Content-Type:Accept-Ranges:Apache/2.2.8 (Ubuntu) DAV/2
```

Based on the preceding result, we know that the target is a `Linux 2.6` machine.

The following screenshot shows the information from the target machine:

```
root@metasploitable:/home/msfadmin# uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 G
NU/Linux
```

By comparing this information, we know that `p0f` got the OS information correctly. The remote machine is using Linux Version 2.6.

You can stop `p0f` by pressing the *Ctrl* + *C* key combination.

Nmap

Nmap is a very popular and capable port scanner. Besides this, it can also be used to fingerprint a remote machine's operating system. It is an active fingerprinting tool. To use this feature, you can give the `-O` option to the `nmap` command.

For example, if we want to fingerprint the operating system used on the `192.168.56.102` machine, we use the following command:

```
nmap -O 192.168.56.102
```

The following screenshot shows the result of this command:

```
MAC Address: 08:00:27:43:15:18 (Cadmus Computer Systems)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.35 seconds
```

Nmap was able to get the correct operating system information after fingerprinting the operating system of a remote machine.

We will talk more about Nmap in a later chapter.